

PARTIAL LEAST SQUARE - PATH MODELING

Metodologia, Software e Aplicação

por

Francesco Costigliola

Trabalho de projecto apresentado como
requisito parcial para obtenção do grau de

Mestre em Estatística e Gestão de Informação

pelo

Instituto Superior de Estatística e Gestão de Informação
da
Universidade Nova de Lisboa

PARTIAL LEAST SQUARE - PATH MODELING

Metodologia, Software e Aplicação

por

Francesco Costigliola

sob orientação de

Professor Doutor Pedro Simões Coelho

Trabalho de projecto apresentado como
requisito parcial para obtenção do grau de

Mestre em Estatística e Gestão de Informação

pelo

Instituto Superior de Estatística e Gestão de Informação

da

Universidade Nova de Lisboa

Lisboa, Novembro de 2009

Ai miei genitori e
a mio fratello Andrea

Conteúdo

Lista das Figuras	II
1 Introdução	1
2 O Algoritmo PLS-PM	6
2.1 Enquadramento Geral	6
2.2 O Modelo PLS-PM	10
2.2.1 O Modelo de Medida	12
2.2.2 O Modelo Estrutural	20
2.3 Estrutura do Algoritmo PLS-PM	21
2.3.1 Estimação Externa	22
2.3.2 Estimação dos Pesos	23
2.3.3 Estimação Interna	25
2.3.4 Processo Iterativo do Algoritmo PLS-PM	27
2.4 Medidas de Fiabilidade e de Validade do Modelo	29

2.4.1	Comunalidade e Redundância	29
2.4.2	A Re-amostragem: <i>Jack-knife</i> e <i>Bootstrap</i>	32
3	Desenvolvimento do Software	34
3.1	Enquadramento Geral	34
3.2	Estrutura do Programa	36
3.3	Informação de Arranque do Algoritmo	40
3.4	<i>Outputs</i> Desenvolvidos	44
4	Aplicação a Dados Reais	49
4.1	Introdução	49
4.2	Os Dados Iniciais	49
4.3	Os Resultados	51
4.4	Processo de Comparação	55
4.4.1	O SmartPLS	55
4.4.2	O SmartPLS: os Resultados	56
4.4.3	Comparação dos Resultados	56
5	Síntese e Conclusão	60

Lista de Figuras

2.1	Notação Gráfica do modelo PLS-PM	11
2.2	O modelo PLS-PM	12
2.3	O Modo Reflectivo	13
2.4	O Modo Formativo	18
2.5	<i>Multiple Indicators for Multiple Causes</i> (MIMIC)	20
2.6	Processo iterativo do algoritmo PLS-PM	28
3.1	Informação de Arranque do Algoritmo: o modelo de exemplo .	40
3.2	Informação de Arranque do Algoritmo: os dados de <i>input</i> (Modo Normal)	42
3.3	Informação de Arranque do Algoritmo: os dados de <i>input</i> (Modo Simulação)	43
3.4	Informação de Arranque do Algoritmo: o ficheiro “sim_n.xls” .	43
3.5	Informação de Arranque do Algoritmo: o ficheiro “model.xls” .	44
3.6	Informação de Arranque do Algoritmo: o ficheiro “inner_design.xls”	44

4.1	Modelo de Satisfação com os Operadores de Rede Móvel . . .	50
4.2	Modelo Estrutural	51
4.3	Modelo de Medida	51
4.4	O Ficheiro de <i>Output</i>	52
4.5	<i>Loadings</i> do Modelo de Medida	52
4.6	Coefficientes das relações entre as variáveis latentes	53
4.7	Pesos Externos	53
4.8	Correlações entre as Variáveis Latentes	54
4.9	Medidas de validade e fiabilidade dos modelos	54
4.10	SmartPLS: <i>Loadings</i> do Modelo de Medida	57
4.11	SmartPLS: Coeficientes das relações entre as variáveis latentes	57
4.12	SmartPLS: Pesos Externos	58
4.13	SmartPLS: Correlações entre as Variáveis Latentes	58
4.14	SmartPLS: Medidas de validade e fiabilidade dos modelos . . .	58

Resumo

O trabalho proposto enquadra-se no âmbito da estatística computacional, tendo como objectivo primário o desenvolvimento de um software estatístico para a implementação do algoritmo PLS - PM (Partial Least Square - Path Modeling). O algoritmo foi programado na sua totalidade em SAS-IML (SAS - Interactive Matrix Language) e complementado com dois módulos de simulação para a estimação da significância dos parâmetros do modelo.

O trabalho apresenta uma parte metodológica baseada na revisão da literatura relevante no âmbito do PLS - PM. De seguida foram apresentadas todas as fases de análise e implementação do software em questão.

Foram indicados sucessivos testes realizados a um conjunto de dados, utilizando o software desenvolvido e outro presente no mercado, de modo a comparar o desempenho de cada um e os resultados obtidos. A comparação dos *outputs* originou resultados positivos, isto é, os resultados produzidos dos dois softwares são iguais.

Além dos resultados de base associados ao PLS-PM, o software criado disponibiliza uma nova técnica de simulação que permite adequar o cálculo da significância às exigências do investigador. Outra vantagem que o programa traz é a possibilidade de executar uma metodologia(PLS-PM) que não está implementada no ambiente SAS.

Este projecto apresenta a particularidade de poder ser potenciado, tornando-se uma ferramenta de arranque para desenvolvimentos futuros devido à circunstância de ter sido desenvolvido e livremente disponibilizado para a comunidade científica.

Abstract

The following work was developed in the context of computational statistics discipline and its main goal is the creation of a statistic software for the implementation of PLS - PM (Partial Least Square - Path Modeling) algorithm. The algorithm has been fully programmed in SAS-IML (SAS - Interactive Matrix Language) and it contains two different simulation techniques that allow the evaluation and determination of parameters significance.

This thesis presents a methodological chapter based on the most relevant literature on PLS - PM and all the phases of analysis and implementation of the mentioned software.

Some tests were applied to a set of data, using the custom-built software and other software available on the market. Those tests were made to compare and analyze the performance and results of the different solutions. The output comparison between two different programs produced positive results because their output was equivalent.

Moreover the software provides a new simulation method which allows to fit the significance calculation to the researcher demands. It also presents the possibility to execute a PLS-PM model, a technique not available in SAS.

Another purpose of this project is the prospect of being further enhanced and become a starting point for future developments. This is possible because this work has been developed and made available in open source within the scientific community.

Capítulo 1

Introdução

A análise de dados é hoje uma metodologia estatística consolidada, que tem como objectivo a análise de fenómenos reais complexos, através da observação conjunta de diversas variáveis associadas ao fenómeno em estudo e das relações existentes entre as mesmas. Para a interpretação da complexidade existente entre as variáveis observadas, é com frequência necessário estruturar um modelo, de modo a estimar as relações que existem entre as mesmas. Os modelos estatísticos têm como objectivo principal descrever, explicar, estimar, reproduzir e investigar fenómenos reais e representam uma exemplificação do mundo real. De entre as diferentes técnicas existentes para a análise das relações entre as variáveis de um sistema, realçam-se os Modelos de Equações Estruturais (*Structural Equation Models* - SEM) que representam um ponto comum entre *Path Analysis* (PA) (Tukey 1964) e *Confirmatory*

Factor Analysis (CFA) (Thurstone 1931).

Os modelos de Equações Estruturais foram introduzidos por Jöreskog em 1970, faziam parte do grupo de modelos confirmatórios e tinham o objectivo de avaliar as relações de causa efeito existentes entre duas ou mais variáveis através do método de estimação da máxima verosimilhança (*Maximum Likelihood*, ML); este método é designado por SEM-ML ou LISREL¹. Ao longo dos anos este foi o único método de estimação para os modelos de Equações Estruturais. O primeiro software associado a estes modelos foi LISREL, desenvolvido por Jöreskog e Sörbom em 1996.

Sucessivamente, foram desenvolvidas muitas outras técnicas de estimação, como GLS (*Generalized Least Squares*) e ADF (*Asymptotically Distribution Free*). Todas as técnicas enunciadas fazem parte do grupo dos métodos *covariance-based*².

O PLS-PM nasceu em 1975 com o objectivo de tornar mais flexível a utilização destes métodos; de facto, foram anuladas as hipóteses formuladas nos métodos *covariance-based*. Nos anos seguintes, foram inúmeras as variantes formalizadas para os métodos *model-based* e inúmeros foram também, sobretudo nos últimos anos, os softwares apresentados.

Hoje, a análise PLS-PM é reconhecida em diversas áreas. Os estudos qui-

¹O LISREL é o primeiro software disponibilizado para a análise dos modelos SEM-ML, tendo-se vindo a afirmar como sinónimo desses modelos de equações estruturais.

²Os métodos *covariance-based* efectuem a estimação dos parâmetros do modelo através da minimização das diferenças entre as matrizes de covariância estimada e da amostra.

miométricos (Jiang, Berry, Siesler & Ozaki 2002), por exemplo, representam o primeiro sector onde o PLS - PM foi aplicado. Hoje em dia esta técnica encontra a sua natural colocação no campo socioeconómico. De facto, é muitas vezes utilizada para a avaliação da satisfação com produtos e/ou serviços, através da análise das relações causais entre variáveis latentes (Fornell, Johnson, Anderson, Cha & Bryant 2002).

Apesar de ter nascido em 1975 com Wold, o PLS-PM está, ainda hoje, no centro de investigações que têm como objectivo a melhoria e diversificação da técnica.

Este projecto apresenta uma primeira secção estritamente teórica, dedicada à definição do modelo PLS-PM e das suas diversas componentes; uma segunda parte que está totalmente focada na estruturação e no desenvolvimento do algoritmo associado ao método definido na primeira secção.

Para a realização deste projecto foi utilizado o software estatístico SAS, mais especificamente o módulo IML (*Interactive Matrix Language*). De entre os vários softwares, foi escolhido o SAS porque não possui nenhum procedimento interno para este tipo de análise (PLS-PM). Dentro do SAS foi seleccionado o módulo IML porque era o único que permitia uma execução otimizada do algoritmo, sendo que, através deste módulo, foi possível programar numa linguagem matricial baseada na álgebra linear.

Este projecto apresenta a particularidade de poder ser potenciado, tornando-

se uma ferramenta de arranque para desenvolvimentos futuros. Esta característica deve-se à circunstância de ter sido desenvolvido e livremente disponibilizado para a comunidade científica. Nessa perspectiva, representa um ponto de chegada e um ponto de partida.

Mais concretamente, foi preocupação central a de contemplar os seguintes elementos:

- Introdução dos dados iniciais em Excel;
- Optimização de algumas partes do algoritmo computacionalmente pesadas;
- Exportação dos *outputs* analíticos em Excel.

Relativamente à estrutura do trabalho, esta apresenta-se da seguinte forma:

- O segundo capítulo serve de introdução ao algoritmo PLS-PM. Nesta secção será apresentada a estimação dos parâmetros do modelo e as estimações do modelo externo e do modelo interno;
- No terceiro capítulo será formalizado o algoritmo de cálculo do PLS-PM, tendo como objectivo principal a apresentação do software;
- No quarto capítulo será apresentada a aplicação do algoritmo desenvolvido com dados reais. Os resultados serão sucessivamente comparados com os resultados produzidos para outro software presente no mercado;

- A última secção da tese será dedicada à síntese, às conclusões e aos possíveis desenvolvimentos futuros do projecto.

Capítulo 2

O Algoritmo PLS-PM

2.1 Enquadramento Geral

Dos Modelos de Equações Estruturais ao PLS-PM

Os modelos de equações estruturais (SEM) incluem diversas técnicas estatísticas que permitem a estimação de relações causais, definidas com base num modelo teórico, através das ligações existentes entre dois ou mais conceitos latentes. Estes conceitos são medidos através de um ou mais indicadores observáveis.

A ideia que subjaz a estes modelos é a de que a complexidade existente num determinado sistema pode ser analisada com base num conjunto de re-

lações causais entre conceitos latentes denominados Variáveis Latentes (VL), cada uma medida através de indicadores definidos como Variáveis Manifestas (VM).

Os modelos de equações estruturais representam um ponto comum entre a *path analysis* (PA) e a análise factorial confirmatória (CFA) (Trinchera 2007). O ponto comum entre os SEM e a CFA é relativo ao significado dos blocos de variáveis; de facto, em ambas as técnicas, cada bloco ou subconjunto de variáveis é considerado como sendo expressão de um único conceito.

A *Path Analysis* tem como objectivo a formulação de modelos relacionais com efeitos directos e indirectos entre variáveis observadas. Quando as variáveis presentes no modelo relacional são latentes e são medidas através de um conjunto de indicadores observados, a *Path Analysis* é definida por Modelos de Equações Estruturais.

As técnicas que se enquadram dentro dos modelos de equações estruturais, podem dividir-se em dois grupos: *Covariance Based* e *Component Based*.

Nos modelos *Covariance Based* os métodos de estimação utilizados são, entre outros: *Maximum Likelihood*, *Generalized Least Squares* e *Asymptotically Distribution Free*. As técnicas que utilizam estes métodos de estimação têm como objectivo a reprodução da matriz de covariância das variáveis manifestas. A hipótese fundamental subjacente a estas abordagens é que a matriz de covariância seja função dos parâmetros do modelo. Esta abordagem é

também definida como *hard modeling*.

Por outro lado, foram desenvolvidas outras técnicas menos exigentes em termos de pressupostos estatísticos. Esta abordagem é designada como *Component Based* (ou *soft modeling*). O PLS-PM enquadra-se neste conjunto de técnicas, utilizando o *Partial Least Squares* como método de estimação. Estas técnicas visam a estimação dos parâmetros através da minimização da variância.

Enquadramento Histórico

A evolução histórica da abordagem PLS (Partial Least Squares) teve início em 1966, quando Herman Wold formalizou a ideia de PLS num artigo dedicado à Análise em Componentes Principais (Wold 1966). Neste artigo foi introduzido o algoritmo NILES (*Nonlinear Iterative LEast Squares*).

A partir de 1973, o mesmo autor estendeu esta técnica à análise das correlações canónicas e o algoritmo ficou conhecido por NIPALS (*Nonlinear Iterative Partial Least Squares*) (Wold 1973, Wold 1975).

A primeira versão finalizada do algoritmo PLS aplicado às variáveis latentes foi apresentada em 1979. As principais referências bibliográficas relativas a esta abordagem são (Jöreskog & Wold 1982, Wold 1982).

Com esta nova abordagem, Wold conseguiu fazer oposição ao SEM-ML (Jöreskog

1970) *hard modeling*¹ através do PLS *soft modeling*². Em 1982 foi apresentado um artigo com a comparação entre as abordagens SEM-ML e PLS (Jöreskog & Wold 1982).

Nos anos seguintes, a abordagem PLS foi utilizada para formalizar a regressão PLS (Wold, Martens & Wold 1983), tendo esta técnica tido imenso sucesso no campo quimiométrico. Ambas as técnicas (PLS para os *path models* e regressão PLS) são designadas como Abordagem PLS. Para evitar confusões entre ambas, Tenenhaus, Vinzi, Chatelin e Lauro (Tenenhaus, Esposito Vinzi, Chatelin & Lauro 2005) seguiram uma sugestão de H. Martens e decidiram utilizar o termo PLS *Path Modeling* para definir o PLS para modelos de equações estruturais.

Seguidamente à formalização da técnica, foram poucas as tentativas de programação de softwares para os modelos de equações estruturais. De facto, durante muitos anos, o único software disponível para estes modelos foi o LVPLS 1.8 desenvolvido por Lohmöller em 1987 (data da última versão disponibilizada). Para encontrar outro software foi preciso esperar até 1989, ano em que Sellin (Sellin 1989) desenvolveu o PLSPATH.

A situação alterou-se radicalmente nos últimos tempos. Hoje, os investigadores desta área têm muitas opções diferentes na escolha do software, tendo

¹São definidas *hard modeling* as técnicas que apresentam fortes hipóteses acerca da distribuição.

²Caracterizada por apresentar poucas hipóteses acerca da distribuição das variáveis em análise.

com o passar dos anos melhorado a facilidade de utilização. O facto dos softwares serem mais *user friendly* permitiu a evolução desta técnica em diversas áreas, como por exemplo no marketing e na investigação organizacional (Diamantopoulos & Winklhofer 2001).

Alguns dos principais softwares que se encontram actualmente no mercado são: PLS-GUI (Li 2005), VisualPLS (Fu 2006), PLS-Graph (Chin 2001), SPAD-PLS (Test & Go 2006) e SmartPLS (Ringle, Wende & Will 2005). Na maioria destes softwares, já são contempladas algumas características mais avançadas, como por exemplo: tratamento de dados em falta, re-amostragem, validação cruzada, etc.

Uma óptima referência para a comparação dos diversos softwares disponíveis é (Temme, Kreis & Hildebrandt 2005).

2.2 O Modelo PLS-PM

Antes de apresentar a metodologia relativa ao PLS-PM, é necessário introduzir algumas notações gráficas (Figura 2.1). Os objectos gráficos presentes na representação dos modelos PLS-PM são:

- As elipses (ou os círculos) são utilizados para as variáveis latentes;
- Os rectângulos (ou os quadrados) são utilizados para as variáveis manifestas;

- As setas representam as relações causais entre as variáveis (latentes e/ou manifestas), sendo que a direcção das mesmas define a direcção da relação (por exemplo, dada uma específica relação entre duas variáveis, a variável que recebe a seta é considerada como sendo a variável endógena).

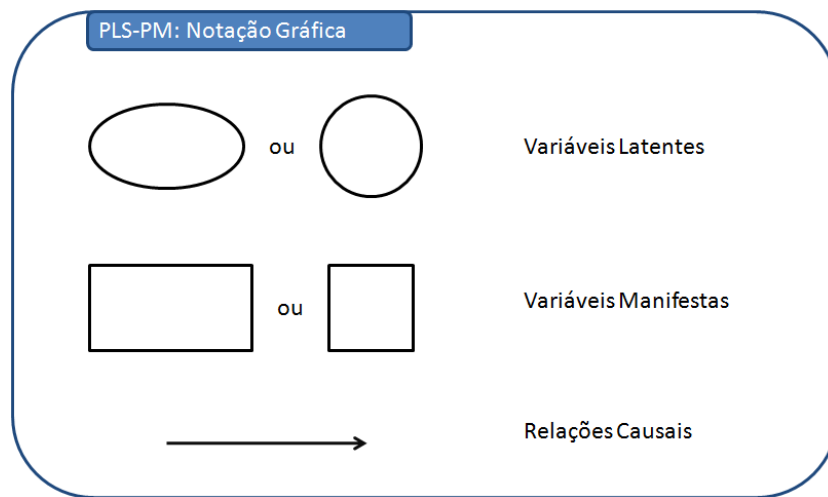


Figura 2.1: Notação Gráfica do modelo PLS-PM

No modelo destacam-se dois sub-modelos: o modelo de medida (ou *outer model*) e o modelo estrutural (ou *inner model*). O primeiro faz referência às relações que existem entre os indicadores e as variáveis latentes, o segundo é relativo às ligações existentes entre as variáveis latentes em estudo (Figura 2.2).

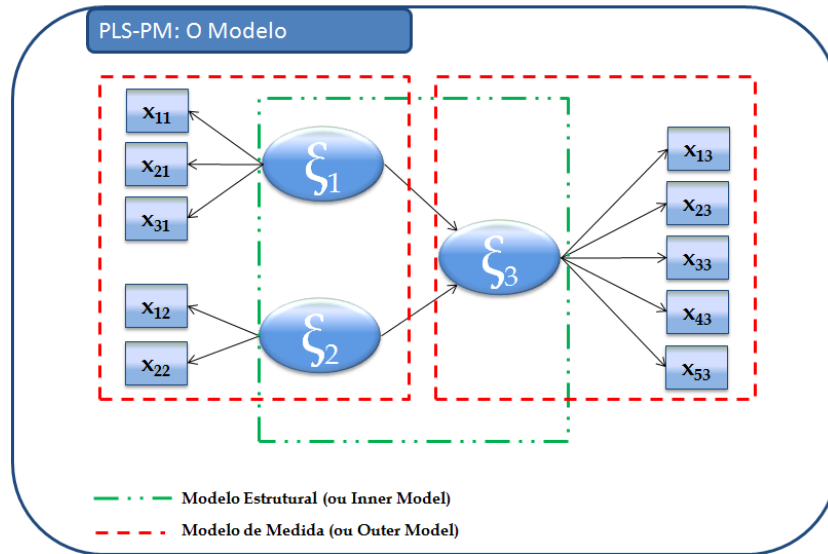


Figura 2.2: O modelo PLS-PM

2.2.1 O Modelo de Medida

Uma variável latente ξ é uma variável (ou conceito) inobservável, descrita de forma indirecta por um bloco de variáveis observadas x_h definidas como variáveis manifestas (ou indicadores). Existem três formas de relacionar as variáveis manifestas às suas variáveis latentes: Modo Formativo, Modo Reflectivo e MIMIC (*Multiple Indicators and Multiple Causes*).

Modo Reflectivo

Nos modelos que utilizam o modo reflectivo, cada variável manifesta está relacionada com a sua variável latente através de uma regressão simples (Figura 2.3).

$$x_h = \pi_{h0} + \pi_h \xi + \epsilon_h \quad (2.1)$$

onde ξ tem média m e desvio padrão igual a 1. Neste esquema de relacionamento, cada variável manifesta x_h reflecte a sua variável latente ξ .

A equação 2.1 tem de obedecer a uma hipótese definida *predictor specification* por H. Wold:

$$E(x_h|\xi) = \pi_{h0} + \pi_h \xi \quad (2.2)$$

Esta hipótese implica que o resíduo ϵ_h tem média igual a 0 e não está correlacionado com a variável ξ .

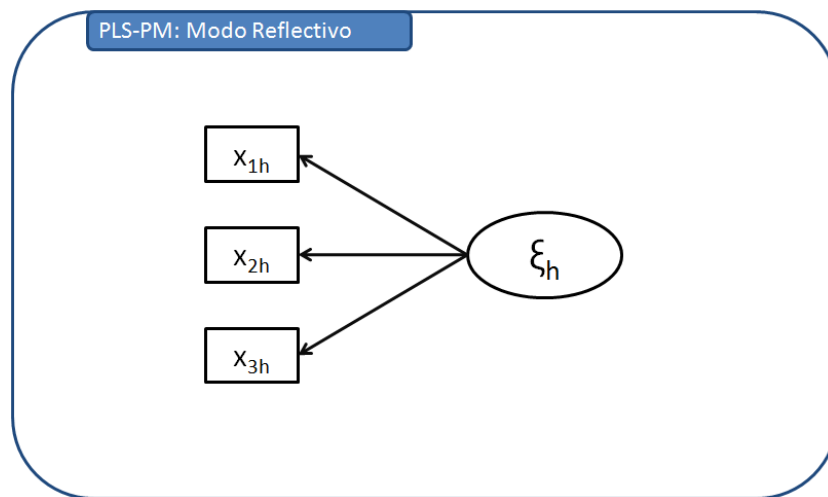


Figura 2.3: O Modo Reflectivo

No modo reflectivo, o bloco de variáveis manifestas tem que ser unidimensio-

nal no sentido da análise factorial. O objectivo é que cada bloco de variáveis manifestas tem que conter variáveis fortemente correlacionadas.

Quando se trabalha com dados reais, é importantíssimo verificar a unidimensionalidade do bloco de variáveis manifestas.

Para o controlo da unidimensionalidade existem três técnicas: a análise em componentes principais do bloco de variáveis manifestas, o α de Cronbach e o ρ de Dillon-Goldstein.

- *Análise em Componentes Principais*: esta técnica permite controlar a unidimensionalidade através da análise dos valores próprios. Se o primeiro valor próprio for superior a 1 e os outros inferiores ou no limite muito afastados do primeiro, podemos considerar o bloco unidimensional. Após controlo do valor próprio, é preciso controlar se todas as variáveis do bloco estão positivamente correlacionadas com o primeiro factor. No caso em que a correlação entre uma das variáveis manifestas e o factor seja negativa, significa que a variável é inadequada para a medição da variável latente à qual está associada. Quando se verifica esta situação é aconselhável remover a variável do bloco de indicadores.
- *α de Cronbach*: esta estatística pode ser utilizada para verificar a unidimensionalidade de um bloco de p variáveis manifestas quando estão positivamente correlacionadas entre si.

Cronbach propôs o seguinte procedimento para variáveis estandardizadas.

A variância do $\sum_{h=1}^p x_h$ está desenvolvida da seguinte maneira:

$$Var\left(\sum_{h=1}^p x_h\right) = p + \sum_{h \neq h'} corr(x_h, x_{h'}) \quad (2.3)$$

Podemos afirmar que o bloco será mais unidimensional quanto maior for $\sum_{h \neq h'} corr(x_h, x_{h'})$.

Com base na equação 2.3 podemos passar ao cálculo do seguinte rácio:

$$\alpha' = \frac{\sum_{h \neq h'} corr(x_h, x_{h'})}{p + \sum_{h \neq h'} corr(x_h, x_{h'})} \quad (2.4)$$

O valor máximo que α' pode atingir é $(p-1)p$ quando todas as $corr(x_h, x_{h'})$ são iguais a 1.

Assim, o α de Cronbach é obtido dividindo o α' pelo seu valor máximo:

$$\alpha = \frac{\sum_{h \neq h'} corr(x_h, x_{h'})}{p + \sum_{h \neq h'} corr(x_h, x_{h'})} \times \frac{p}{p-1} \quad (2.5)$$

Para as variáveis originais o α calcula-se da seguinte forma:

$$\alpha = \frac{\sum_{h \neq h'} corr(x_h, x_{h'})}{Var(\sum_{h=1}^p x_h)} \times \frac{p}{p-1} \quad (2.6)$$

Um bloco é considerado unidimensional quando o α é maior de 0.7.

- ρ de Dillon-Goldstein: Por construção, o sinal da correlação entre as variáveis manifestas x_h e a variável latente ξ tem que ser positivo, isto significa que na equação 2.1 todos os loadings π_h são positivos. Um bloco pode-se definir unidimensional quanto maiores forem os loadings. O primeiro passo refere-se sempre ao cálculo da variância de $\sum_{h=1}^p x_h$, mas desta vez é determinada a partir da equação 2.1 supondo que os termos residuais ϵ_h são independentes:

$$\begin{aligned} Var \left(\sum_{h=1}^p x_h \right) &= Var \left(\sum_{h=1}^p (\pi_{h0} + \pi_h \xi + \epsilon_h) \right) \\ &= \left(\sum_{h=1}^p \pi_h \right)^2 Var(\xi) + \sum_{h=1}^p Var(\epsilon_h) \end{aligned} \quad (2.7)$$

Um bloco resulta mais unidimensional quanto maior for $(\sum_{h=1}^p \pi_h)^2$.

O ρ de Dillon-Goldstein é determinado como:

$$\rho = \frac{(\sum_{h=1}^p \pi_h)^2 Var(\xi)}{(\sum_{h=1}^p \pi_h)^2 Var(\xi) + \sum_{h=1}^p Var(\epsilon_h)} \quad (2.8)$$

Dado que nesta fase não possuímos os valores de ξ , será necessário estimar o seu valor.

Supondo que todas as variáveis manifestas x_h e a variável latente ξ são estandardizadas, podemos obter uma aproximação da variável latente utilizando o primeiro factor da análise em componentes principais t_1 relativa ao bloco das variáveis manifestas.

O loading π_h pode ser estimado através da $\text{corr}(x_h, t_1)$ e, sempre com base na equação 2.1, a $\text{Var}(\epsilon_h)$ é estimada através de $1 - \text{corr}^2(x_h, t_1)$.

O ρ de Dillon-Goldstein estimado será igual a:

$$\hat{\rho} = \frac{[\sum_{h=1}^p \text{corr}(x_h, t_1)]^2}{[\sum_{h=1}^p \text{corr}(x_h, t_1)]^2 + \sum_{h=1}^p [1 - \text{corr}^2(x_h, t_1)]} \quad (2.9)$$

Um bloco é unidimensional quando o $\hat{\rho}$ de Dillon-Goldstein é maior do que 0.7.

Segundo Chin (Chin 1998), esta estatística é considerada a melhor para a verificação da unidimensionalidade de um bloco de variáveis manifestas.

Modo Formativo

No modo formativo, a variável latente ξ é obtida através de uma combinação linear das variáveis manifestas (Figura 2.4) a ela associadas mais um termo residual:

$$\xi = \sum_{h=1}^p \varpi_h x_h + \delta \quad (2.10)$$

Neste esquema, um bloco de variáveis manifestas pode ser multidimensional.

A hipótese relativa ao *predictor specification* para a equação 2.10 é:

$$E(\xi | x_1, \dots, x_p) = \sum_{h=1}^p \varpi_h x_h \quad (2.11)$$

Esta hipótese implica que o vector dos resíduos δ tem média 0 e não está correlacionado com as variáveis manifestas x_h .

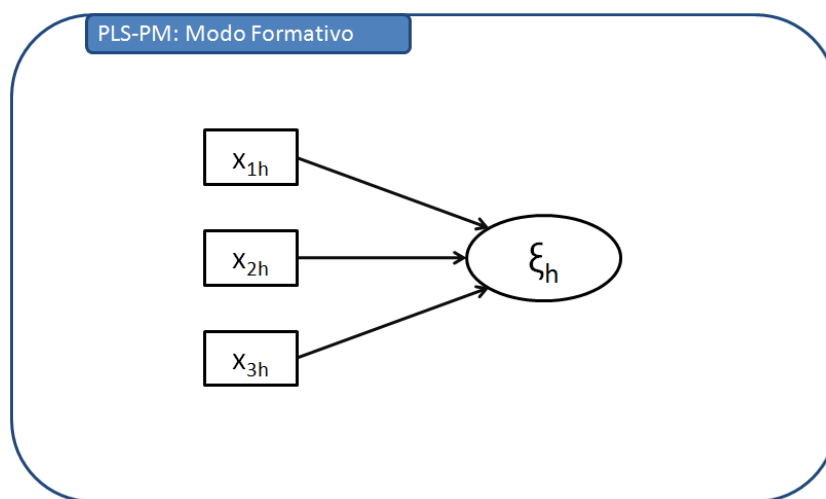


Figura 2.4: O Modo Formativo

Neste esquema, a variável latente é gerada a partir da combinação linear das variáveis manifestas e o sinal de cada peso ϖ_h é, com frequência calculado com base no sinal da correlação existente entre a variável manifesta x_h e a variável

latente ξ . No algoritmo clássico não existem vínculos para a definição dos loadings ou dos pesos, mas existe a possibilidade de adicioná-los (Tenenhaus et al. 2005).

MIMIC (*Multiple Indicators for Multiple Causes*)

O esquema MIMIC é uma fusão entre o esquema reflectivo e o esquema formativo (Figura 2.5).

O modelo de medida para um determinado bloco é:

$$x_h = \pi_{h0} + \pi_h \xi + \epsilon_h \quad (2.12)$$

onde $h \in P_1$.

A variável latente ξ é definida como:

$$\xi = \sum_{h \notin P_1} \varpi_h x_h + \delta \quad (2.13)$$

onde P_1 é o conjunto das variáveis manifestas que seguem o esquema reflectivo.

A hipótese do *predictor specification* para as equações 2.12 e 2.13 são as mesmas definidas respectivamente nos esquemas reflectivo e formativo.

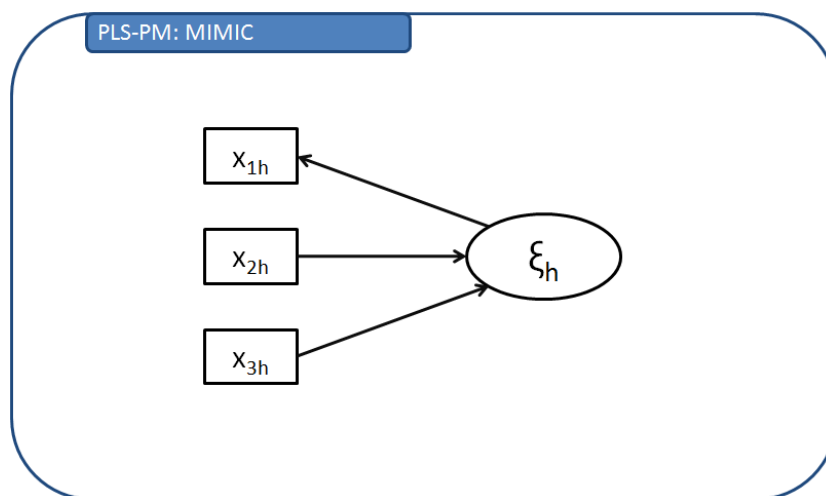


Figura 2.5: *Multiple Indicators for Multiple Causes* (MIMIC)

2.2.2 O Modelo Estrutural

O modelo causal da figura 2.2 apresenta as variáveis latentes relacionadas entre si. Estas relações compõem o modelo estrutural (ou *inner model*) e podem ser formalizadas da seguinte forma:

$$\xi_j = \beta_{j0} + \sum_i \beta_{ji} \xi_i + \nu_j \quad (2.14)$$

Para a equação 2.14 também está definida a hipótese do *predictor specification*.

As variáveis latentes que nunca aparecem como dependentes na equação 2.14 são definidas variáveis exógenas; as que aparecem pelo menos uma vez como dependentes são definidas variáveis endógenas.

O modelo causal representa uma cadeia causal, isto é, não existem ciclos. Estes tipos de modelos são definidos como recursivos do Latim *Recurzio*, que significa *Eu posso voltar* (Tenenhaus et al. 2005).

Um modelo estrutural pode ser descrito através de uma matriz triangular inferior, de dimensão igual ao número de variáveis latentes que pode apresentar zeros ou uns. Lohmöller define esta matriz como *inner design matrix*.

A célula (i, j) estará preenchida com um 1 quando a variável latente j explica a variável i ; caso contrário será igual a 0.

2.3 Estrutura do Algoritmo PLS-PM

De entre os principais algoritmos para os modelos de equações estruturais, destacam-se o algoritmo desenvolvido por Herman Wold (Jöreskog & Wold 1982, Wold 1982) e o algoritmo de Lohmöller (Lohmöller 1987, Lohmöller 1989).

Para o desenvolvimento deste projecto foi utilizado o algoritmo original de H. Wold, que será apresentado em detalhe nos próximos parágrafos.

A estimação das variáveis latentes ξ_j é feita com base nos seguintes passos que serão iterados até alcançar a convergência dos pesos:

1. Estimação Externa;
2. Estimação Interna;

3. Estimação dos Pesos;

2.3.1 Estimação Externa

A primeira fase tem como objectivo a estimação dos y_j . As variáveis latentes estandardizadas (média = 0 e desvio padrão = 1) são estimadas como combinação linear das suas variáveis manifestas centradas:

$$y_j \propto \pm \left[\sum w_{jh} + (x_{jh} - \bar{x}_{jh}) \right] \quad (2.15)$$

onde o símbolo \propto significa que a variável da esquerda é proporcional ao operador da direita e o sinal \pm representa a ambiguidade dos sinais a utilizar. Este problema é resolvido seleccionando o sinal que faz com que a variável y_j esteja correlacionada positivamente com a maioria das variáveis manifestas x_{jh} .

A j -ésima variável latente estimada é calculada da seguinte forma:

$$y_j = \sum \tilde{w}_{jh} + (x_{jh} - \bar{x}_{jh}) \quad (2.16)$$

Os coeficientes w_{jh} e \tilde{w}_{jh} são denominados *outer weights* (pesos externos).

A média m_j é estimada da seguinte maneira:

$$\hat{m}_j = \sum \tilde{w}_{jh} \bar{x}_{jh} \quad (2.17)$$

e a variável latente ξ_j é estimada como segue

$$\hat{\xi}_j = \sum \tilde{w}_{jh} x_{jh} = y_j + \hat{m}_j \quad (2.18)$$

A estimação dos pesos \tilde{w}_{jh} está apresentada no seguinte parágrafo.

2.3.2 Estimação dos Pesos

Existem duas maneiras diferentes de calcular os pesos w_{jh} : modo A e modo B.

Modo A

No modo A, os pesos w_{jh} representam o coeficiente de regressão de z_j na regressão simples de x_{jh} sobre z_j :

$$w_{jh} = cov(x_{jh}, z_j) \quad (2.19)$$

onde a variável z_j é estandardizada.

Modo B

No modo B, o vector dos pesos w_{jh} representa o vector dos coeficientes de regressão na regressão múltipla de z_j sobre as variáveis manifestas centradas $(x_{jh} - \bar{x}_{jh})$ relacionadas com a variável latente ξ_j :

$$w_j = \left(X_j' X_j \right)^{-1} X_j' z_j \quad (2.20)$$

onde X_j é a matriz que tem por coluna as variáveis manifestas centradas $(x_{jh} - \bar{x}_{jh})$ relacionadas com a mesma variável latente ξ_j .

O modo A é apropriado para os blocos que apresentam um modelo de medida reflectivo e o modo B para os formativos. Para os modelos MIMIC são utilizadas ambos os modos de estimação dos pesos w_{jh} (o modo A para a parte formativa e o modo B para a reflectiva).

O modo B não é utilizado frequentemente por causa da existência de multicolinearidade nos blocos de variáveis manifestas. Nestes casos, em vez das regressões OLS, são utilizadas as regressões PLS.

No caso de se ter *a priori* algum conhecimento relativo à estrutura do modelo, podem-se adicionar vínculos de sinal aos pesos existentes no modelo. O sinal dos pesos pode ser calculado da seguinte forma:

$$\text{sign}(w_{jh}) = \text{sign}(\text{cor}(x_{jh}, \xi_j)) \quad (2.21)$$

Para o modo A, a equação 2.19 pode ser substituídas por

$$\begin{aligned} w_{jh} &= cov(x_{jh}, z_j) & se \ sign[cov(x_{jh}, z_j)] &= sign[cor(x_{jh}, \xi_j)] \\ &= 0 & se \ sign[cov(x_{jh}, z_j)] &\neq sign[cor(x_{jh}, \xi_j)] \end{aligned} \quad (2.22)$$

Para o modo B, a regressão OLS de z_j sobre X_j é substituída pela regressão de z_j sobre X_j tendo em conta o vínculo apresentado na expressão 2.21. Esta regressão múltipla levará a uma regressão onde os coeficientes que apresentam o sinal correcto serão deixados iguais e os que apresentam o sinal contrário ao esperado serão postos iguais a zero.

2.3.3 Estimação Interna

O objectivo do modelo interno, tal como apresentado para o modelo de medida, é a estimação das variáveis latentes. A estimação interna z_j das variáveis latentes standardizadas $(\xi_j - m_j)$ é definida como

$$z_j \propto \sum_{j' : \xi_{j'} \text{ está ligada com } \xi_j} e_{jj'} y_{j'} \quad (2.23)$$

onde os pesos internos $e_{jj'}$ podem ser estimados através da utilização de diversas técnicas.

Uma das técnicas utilizadas com maior frequência é o esquema do centróide (*centroid scheme*) que estima os pesos internos da seguinte forma:

$$e_{jj'} = \text{sign} [\text{cor} (y_j, y_{j'})] \quad (2.24)$$

Neste caso os pesos $e_{jj'}$ são iguais aos sinais da correlação entre y_j e a variável latente $y_{j'}$ relacionada com a y_j (eq. 2.24). Dizem-se relacionadas duas variáveis quando existe uma ligação entre elas, isto é, no modelo causal, existe uma seta que vai de uma variável para a outra descrevendo uma relação de causalidade entre elas.

O esquema do centróide apresenta alguns inconvenientes quando se encontram correlações muito próximas de zero, sendo que estas correlações costumam flutuar à volta do zero, apresentando valores positivos e negativos. Nas aplicações práticas, este problema não representa grandes complicações do ponto de vista das estimações.

Outros esquemas de pesos definidos para a fase de estimação interna são o esquema factorial (*Factorial Scheme*) onde os pesos $e_{jj'}$ são calculados da seguinte forma:

$$e_{jj'} = r_{jj'} = \text{cor} (y_j, y_{j'}) \quad (2.25)$$

e o esquema estrutural (*Structural Scheme*) calculado como segue:

$$\begin{aligned}
 e_{jj'} &= \text{coeficiente de regressão de } y_j \text{ sobre } y_{j'} \text{ se } \xi_{j'} \text{ é explicativa de } \xi_j \\
 &= r_{jj'} \text{ se } \xi_j \text{ é explicativa de } \xi_{j'}
 \end{aligned}
 \tag{2.26}$$

No algoritmo formulado por H. Wold, a estimação interna é representada pelo termo à direita da equação 2.23 e não é efectuada a standardização às variáveis.

Para a execução deste projecto foi utilizado o esquema do centróide e as variáveis foram standardizadas. A standardização efectuada não apresenta nenhuma alteração do ponto de vista da estimação final.

2.3.4 Processo Iterativo do Algoritmo PLS-PM

O primeiro passo do algoritmo PLS-PM baseia-se na definição de um vector de pesos w_{jh} arbitrários para o arranque do mesmo. Os pesos a definir devem estar standardizados, de modo a ser possível obter estimações das variáveis latentes com variância unitária.

Uma das escolhas mais frequentes para os pesos iniciais é $w_{jh} = \text{sign} [\text{cor} (x_{jh}, \xi_j)]$ ou mais simplesmente $w_{jh} = \text{sign} [\text{cor} (x_{jh}, \xi_j)]$ quando $h = 1$ e 0 no caso contrário.

Os passos 2.15, 2.23 e 2.19 ou 2.20, são iterados até alcançar a convergência dos pesos.

Após a execução do último passo, são utilizadas as expressões 2.16, 2.17 e 2.18 para o cálculo de pesos internos \tilde{w}_{jh} , das variáveis latentes estandarizadas $y_j = \sum \tilde{w}_{jh} (x_{jh} - \bar{x}_{jh})$, da media estimada $\hat{m}_j = \sum \tilde{w}_{jh} \bar{x}_{jh}$ das variáveis latentes ξ_j e a estimação final de ξ_j calculada da seguinte forma $\hat{\xi}_j = \sum \tilde{w}_{jh} x_{jh} = y_j + \hat{m}_j$.

O processo iterativo do algoritmo PLS-PM pode ser resumido na seguinte figura:

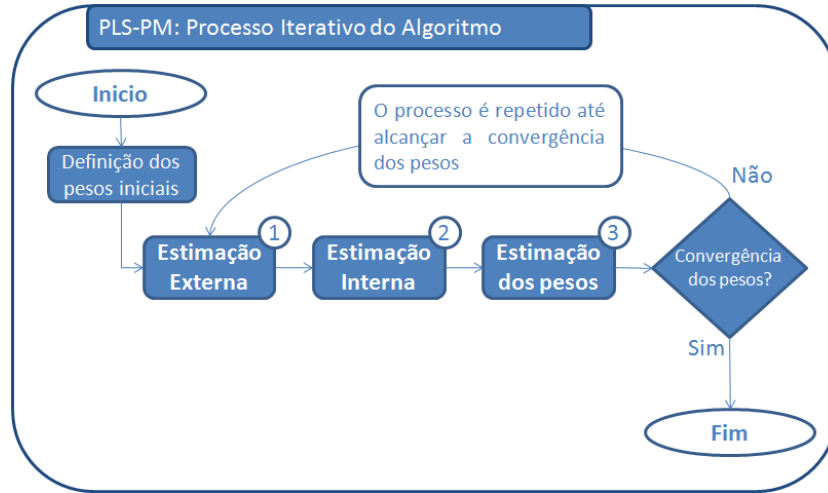


Figura 2.6: Processo iterativo do algoritmo PLS-PM

2.4 Medidas de Fiabilidade e de Validade do Modelo

Um modelo PLS-PM pode ser validado a três níveis diferentes:

- Qualidade do Modelo de Medida;
- Qualidade do Modelo Estrutural;
- Análise das Equações de Regressão Estrutural.

Nos próximos parágrafos serão apresentados os diversos indicadores disponíveis para a validação do modelo.

2.4.1 Comunalidade e Redundância

O índice de comunalidade é utilizado para medir a qualidade do modelo externo relativa a cada bloco.

Para o j -ésimo bloco a comunalidade é

$$comunalidade_j = \frac{\sum_{h=1}^{p_j} cor^2(x_{jh}, y_j)}{p_j} \quad (2.27)$$

A comunalidade média é igual ao valor médio de todas as $cor^2(x_{jh}, y_j)$:

$$\overline{comunalidade} = \frac{\sum_{j=1}^J p_j comunalidade_j}{p} \quad (2.28)$$

onde p é o número total de variáveis manifestas que pertencem a todos os blocos presentes no modelo.

No presente trabalho a comunalidade corresponde ao AVE (Variância Média Extraída) devido à estandardização dos dados. De facto a fórmula da comunalidade (ver equação 2.27) neste caso específico é igual ao AVE (ver equação 2.29).

O AVE³ é a parte da variância das variáveis manifestas de um determinado bloco explicada pela variável latente associada a esse bloco e é calculada da seguinte forma:

$$AVE = \frac{\sum_i \lambda_i^2}{\sum_i \lambda_i^2 + \sum_i Var(\epsilon_i)} \quad (2.29)$$

onde λ_i é o coeficiente da equação do modelo de medida e ϵ_i é o resíduo dessa mesma equação.

O índice de redundância mede a qualidade do modelo estrutural para cada bloco endógeno, tendo em conta o modelo de medida. Mais concretamente, este índice mede a parte de variabilidade das variáveis manifestas ligadas à j -ésima variável latente endógena explicada pela variáveis latentes não directamente ligadas a esta.

Para o bloco j , o índice de redundância é igual a:

³Uma resenha detalhada do AVE encontra-se em (Vilares & Coelho 2005).

$$redundância_j = comunalidade_j \times R^2(y_j, \{todas as y_{j'} \text{ que explicam } y_j\}) \quad (2.30)$$

Para ter uma visão global da redundância é possível calcular o índice de redundância médio, tal como apresentado para o índice de comunalidade:

$$\overline{redundância} = \frac{\sum_{j=1}^J q_j redundância_j}{q} \quad (2.31)$$

onde q é o número total de blocos endógenos presentes no modelo.

Em 2004, foi apresentado um índice de bondade de ajustamento global denominado *GoF* (*Goodness-of-Fit*). O *GoF* é calculado como média geométrica do índice de comunalidade médio ($\overline{comunalidade}$) e do R^2 médio ($\overline{R^2}$):

$$GoF = \sqrt{\overline{comunalidade} \times \overline{R^2}} \quad (2.32)$$

Contrariamente ao que acontece com o SEM-ML, no PLS-PM, o algoritmo não tem como objectivo a optimização de uma função escalar e isto traduz-se na inexistência de um índice de validação global do modelo. O *GoF* representa a solução operacional para este problema, dado que pode ser interpretado como índice de validação global do modelo PLS.

2.4.2 A Re-amostragem: *Jack-knife* e *Bootstrap*

Outro grupo de técnicas que podem ser utilizadas para a validação do modelo são as técnicas de re-amostragem. Estas permitem calcular o grau de significância dos parâmetros estimados pelo modelo.

O *Jack-knife*

O procedimento *Jack-knife* (Chin 1998) baseia-se na construção de amostras apagando um determinado número de observações da amostra inicial (com N observações). O procedimento padrão é o de apagar uma observação de cada vez, obtendo sub-amostras de $N - 1$ observações. Com o aumento do número de observações apagadas, perde-se consistência no cálculo da estatística t devido ao pequeno número de sub-amostras que se conseguem obter.

O *Bootstrap*

O procedimento *Bootstrap* baseia-se na construção de um determinado número de sub-amostras com o mesmo número de observações da amostra inicial. As observações são seleccionadas da amostra inicial através de uma tiragem com reposição. É aconselhável utilizar um número (designado por n) de sub-amostras superior a 100, de forma a obter estimações dos desvios padrão mais razoáveis.

Esta técnica pode ser resumida em duas fases repetidas n vezes:

1. Re-amostragem: esta primeira fase consta na tiragem de uma amostra com reposição com base nas observações iniciais;
2. Distribuição da amostra: nesta fase calculam-se algumas estatísticas dos parâmetros estimados. No presente trabalho foram calculadas a média e o desvio padrão das n com base nas seguintes fórmulas:

$$Mean_{Bootstrap} = \frac{\sum_{i=1}^n \bar{x}_i^*}{n} \quad (2.33)$$

$$StandardDeviation_{Bootstrap} = \sqrt{\frac{\sum_{i=1}^n (\bar{x}_i^* - Mean_{Bootstrap})^2}{n - 1}} \quad (2.34)$$

onde \bar{x}_i^* é o valor do parâmetro estimado na i -ésima iteração do processo *Bootstrap*.

Capítulo 3

Desenvolvimento do Software

3.1 Enquadramento Geral

Introdução

A implementação do algoritmo PLS - PM foi efectuada tendo como objectivo o desenvolvimento de uma ferramenta completa e genérica, ou seja, com capacidade para resolver os mais diversos casos (sem apresentar limitações em termos de variáveis latentes ou manifestas utilizadas, reduzindo a intervenção do investigador na fase de execução do algoritmo), provenientes de áreas de estudo distintas. Nesse sentido, foram analisados os softwares existentes no mercado tendo em vista a implementação dos pontos fortes e desenvolver os

pontos fracos (ou os pontos não desenvolvidos) de cada um.

Um dos pontos críticos será certamente a “velocidade” de execução do algoritmo. Este aspecto está intimamente ligado com a forma como o algoritmo está projectado e optimizado.

Para conseguir um elevado nível de desempenho desenvolveu-se o algoritmo inteiramente no SAS - IML (SAS - *Interactive Matrix Language*) (SASInstitute 2004), dado o elevadíssimo grau de optimização das funções nele contidas. Outra grande vantagem deriva das estruturas de dados utilizadas nesta linguagem, as matrizes; de facto, a linguagem IML baseia-se totalmente nos princípios da álgebra linear tornando muito mais rápidos e intuitivos os cálculos de grandes quantidades de dados.

Os Módulos

Para o desenvolvimento do código foi utilizada uma estrutura de programação presente no IML: os módulos. Estes permitem uma programação mais clara e de fácil interpretação e foram utilizados com o objectivo de gerar blocos de código invocáveis a partir de qualquer parte do programa, isto é, criar uma função ou sub-rotina definidas num “ambiente separado”.

Um módulo pode ser entendido como função ou sub-rotina. Quando o módulo devolve um parâmetro é considerado uma função e é invocado como

uma função interna do IML (*built-in function*) através dos comandos CALL ou RUN. Em todos os outros casos os módulos são entendidos como sub-rotinas. Ao utilizar os módulos como sub-rotinas, a declaração das variáveis cumpre um papel fundamental. De facto, existem variáveis locais e globais: as primeiras só existem no módulo e não podem ser utilizadas no programa que invoca o módulo; as globais, como sugere o nome, podem ser utilizadas dentro e fora do módulo.

Para melhor compreensão dos parágrafos seguintes é apresentado um exemplo da estrutura de um módulo:

```
1  START nome_módulo (variável1 , variável2) GLOBAL(variávelGlobal1);  
2  .  
3  .  
4  .  
5  FINISH nome_módulo;
```

Este módulo apresenta duas variáveis locais (variável1 e variável2) e uma única variável global (variávelGlobal).

3.2 Estrutura do Programa

Nesta secção será apresentada a estrutura do algoritmo implementada no ambiente SAS. O algoritmo implementado, tal como referido no capítulo anterior, é o algoritmo de H. Wold.

O algoritmo foi estruturado da seguinte maneira:

- 1 Importação dos dados e dos ficheiros de suporte
- 2 Estandardização dos dados
- 3 Inicialização do algoritmo
- 4 Estimação do modelo estrutural
- 5 Geração dos pesos externos
- 6 Estimação do modelo de medida
- 7 Estimação do modelo estrutural
- 8 Cálculo dos novos pesos
- 9 Iteração dos passos 6, 7 e 8 do algoritmo até à convergência dos pesos
- 10 Cálculo dos output e das medidas de qualidade
- 11 Cálculo da significância dos parâmetros
- 12 Exportação do ficheiro de resultados

A cada passo corresponde um ou mais módulos.

Nos seguintes parágrafos será apresentada a interacção entre os diversos módulos presentes no programa.

O algoritmo começa com uma parte de código desenvolvida no ambiente SAS/Base relativa à importação dos dados e dos ficheiros necessários para o correcto funcionamento do programa (ver Anexo 1, de linha 1 até 21).

Após importação dos dados, foi efectuada a sua estandardização com base no módulo **stdize** (ver Anexo 1, de linha 45 até 57). A padronização foi efectuada com base no seguinte cálculo:

$$y_i = \frac{(x_i - \mu_x)}{\sigma_x} \quad (3.1)$$

Uma vez padronizados os dados foi inicializado o algoritmo. Esta fase está associada ao módulo **inlize** (ver Anexo 1, de linha 61 até 77) que tem como objectivo a criação de uma matriz com as variáveis latentes do modelo. Dado que cada variável latente é calculada como uma combinação linear das variáveis manifestas e que os pesos externos (outer weights) são iguais a $[1, 0, \dots, 0]$, a variável latente será, nesta primeira iteração, igual à primeira variável manifesta.

Após obtenção das variáveis latentes, foi efectuada a estimação do modelo estrutural (*inner model*) através do módulo **inner_est** (ver Anexo 1, de linha 82 até 101) e sucessivamente foi criado um vector com os pesos externos do modelo através do módulo **create_wgt**. Este módulo tem como único objectivo a criação do vector, sendo que os pesos contidos no mesmo são aqueles utilizados no módulo **inlize** apresentado anteriormente. A estimação do modelo estrutural é efectuada com o esquema do centróide (*centroid scheme*).

A partir deste momento o algoritmo foi baseado num módulo iterativo, o módulo **core** (ver Anexo 1, de linha 213 até 408). Este módulo executa iterativamente três módulos na seguinte ordem:

1. **outer_est** (ver Anexo 1, de linha 106 até 130) que efectua a estimação do modelo de medida (outer model);
2. **inner_est** (ver Anexo 1, de linha 82 até 101) relativo à estimação do modelo estrutural (inner model) apresentado anteriormente;
3. **outer_wgt** (ver Anexo 1, de linha 168 até 190) que calcula os novos pesos externos calculados através de uma regressão linear múltipla.

O critério de paragem do módulo **core** é $(|W_i - W_{i-1}|) < 1.0e - 5$.

Após a convergência do algoritmo foram calculados os *outputs* do modelo e respectivas medidas de qualidade.

O programa contempla dois módulos relativos à técnica de *Bootstrap*. Ambos os módulos têm o objectivo de calcular a significância dos parâmetros estimados do modelo e só diferem no formato dos dados de *input*. O módulo **bootstrap** (ver Anexo 1, de linha 413 até 469) recebe em *input* a base completa e efectua n re-amostragens com reposição; o módulo **bootstrap_sim** (ver Anexo 1, de linha 475 até 551) recebe um ficheiro Excel com n bases que passaram para um processo de re-amostragem externo ao programa.

Todos os *outputs* calculados foram exportados para Excel através de um bloco de código SAS/Base (ver Anexo 1, de linha 597 até 620).

3.3 Informação de Arranque do Algoritmo

Nos parágrafos anteriores foi analisada a importação dos dados e dos ficheiros necessários para o funcionamento do programa. Neste paragrafo serão detalhados os quatro ficheiros a passar em *input*, apresentando o respectivo formato.

Para facilitar a compreensão da estrutura dos ficheiros será utilizado um exemplo que servirá de base para a construção dos mesmos. O exemplo utilizado é:

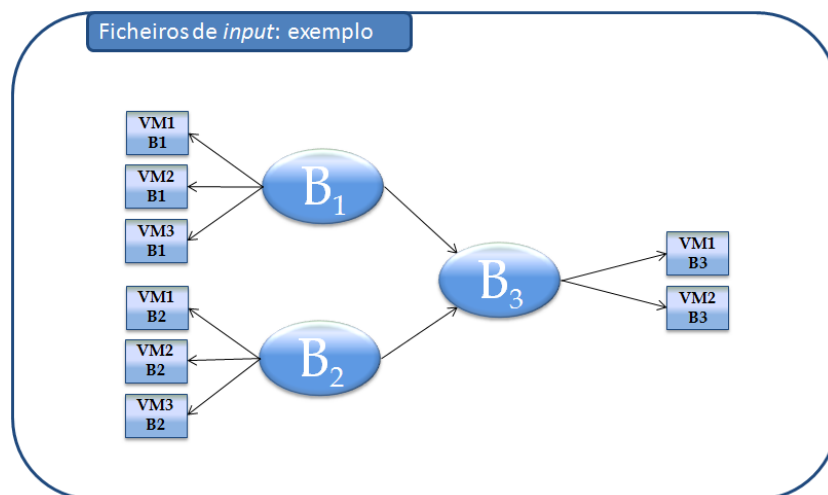


Figura 3.1: Informação de Arranque do Algoritmo: o modelo de exemplo

Todos os ficheiros deverão estar no formato “.xls” (Microsoft Office Excel®).

Os ficheiros necessários para o correcto funcionamento do algoritmo são:

- **dados:** este ficheiro terá que conter os dados originais do modelo (va-

riáveis manifestas não padronizadas). A ordem das variáveis deve ser coerente com a ordem utilizada nos ficheiros que seguem (por exemplo, se as primeiras colunas apresentam as variáveis manifestas pertencentes ao bloco da variável latente “ x ”, no ficheiro que contém a informação relativa às variáveis latentes a variável “ x ” terá que ser a primeira).

O ficheiro de dados pode ter dois formatos diferentes conforme o objetivo pré-fixado. De facto, o programa permite a execução do algoritmo em dois modos: modo normal e modo simulação. A diferença entre ambos é relativa ao processo de cálculo da significância dos parâmetros estimados: o primeiro utiliza o método *Bootstrap* e o segundo utiliza as bases introduzidas no ficheiro de *input* “dados.xls”. Para este último caso, a informação presente no ficheiro de dados deverá ser complementada com a informação contida no ficheiro “sim_n.xls” apresentado nos parágrafos sucessivos.

Portanto, no caso de se querer executar o algoritmo e calcular a significância com o método *Bootstrap*, o ficheiro terá n linhas e p colunas (sendo n o número de observações e p o número de variáveis manifestas do modelo) (ver o gráfico 3.2).

No outro caso, o ficheiro “dados.xls” terá $n_1 + n_2 + n_3 + \dots + n_k$ linhas e p colunas (sendo n_i o número de observações da i -ésima base, p o número

Ficheiros de *input*: dados.xls

	A	B	C	D	E	F	G	H
1	VM1 B1	VM2 B1	VM3 B1	VM1 B2	VM2 B2	VM3 B2	VM1 B3	VM2 B3
2	7	7	6	7	5	5	5	4
3	10	10	9	10	9	10	10	9
4	7	7	7	8	7	6	4	7
5	7	10	5	10	10	5	5	10
6	8	7	10	10	10	5	8	9
7	10	9	7	8	9	10	8	9
8	9	6	2	8	7	1	7	7
9	5	6	1	8	8	9	7	9
10	7	7	7	7	9	6	6	8
11	8	7	3	6	7	7	8	8
12								

Variável Latente **B1** Variável Latente **B2** Variável Latente **B3**

Figura 3.2: Informação de Arranque do Algoritmo: os dados de *input* (Modo Normal)

de variáveis manifestas do modelo e k o número de amostras utilizadas no processo de re-amostragem). No gráfico 3.3 será apresentado um caso com 3 bases com dimensões n_1 , n_2 e n_3 .

- **sim_n**: o ficheiro “sim_n.xls” apenas é utilizado quando se quer calcular a significância dos parâmetros estimados através do modo simulação supra referido. Este ficheiro terá que conter a informação relativa à dimensão das bases introduzidas, tal como se apresenta no seguinte exemplo:
- **model**: o ficheiro “model.xls” contém o número de variáveis manifestas associadas a cada variável latente. A estrutura do ficheiro é:

Ficheiros de *input*: dados.xls

	A	B	C	D	E	F	G	H
1	VM1 B1	VM2 B1	VM3 B1	VM1 B2	VM2 B2	VM3 B2	VM1 B3	VM2 B3
2	7	7	6	7	5	5	5	4
3	10	10	9	10	9	10	10	9
4	7	7	7	8	7	6	4	7
5	7	10	5	10	10	5	5	10
6	10	10	9	10	9	10	10	9
7	9	6	2	8	7	1	7	7
8	7	7	7	8	7	6	4	7
9	8	7	10	10	10	5	8	9
10	10	9	7	8	9	10	8	9
11	5	6	1	8	8	9	7	9
12	7	7	7	7	9	6	6	8
13	8	7	3	6	7	7	8	8

$n_1 = 4$ (rows 2-5)
 $n_2 = 3$ (rows 7-9)
 $n_3 = 5$ (rows 10-13)

Variável Latente **B1** (columns A-C)
 Variável Latente **B2** (columns D-F)
 Variável Latente **B3** (columns G-H)

Figura 3.3: Informação de Arranque do Algoritmo: os dados de *input* (Modo Simulação)

Ficheiros de *input*: sim_n.xls

	A	B	C	D	E	F
1	n					
2	4					
3	3					
4	5					
5						
6						
7						
8						

Número de observações da primeira base (row 2)
 Número de observações da segunda base (row 3)
 Número de observações da terceira base (row 4)

Figura 3.4: Informação de Arranque do Algoritmo: o ficheiro “sim_n.xls”

- **inner_design**: o ficheiro “inner_design.xls” contém a inner design matrix apresentada na secção 2.2.2. As linhas deste ficheiro contêm a informação relativa à variável latente dependente. Neste caso específico, a terceira variável latente depende da primeira e da segunda

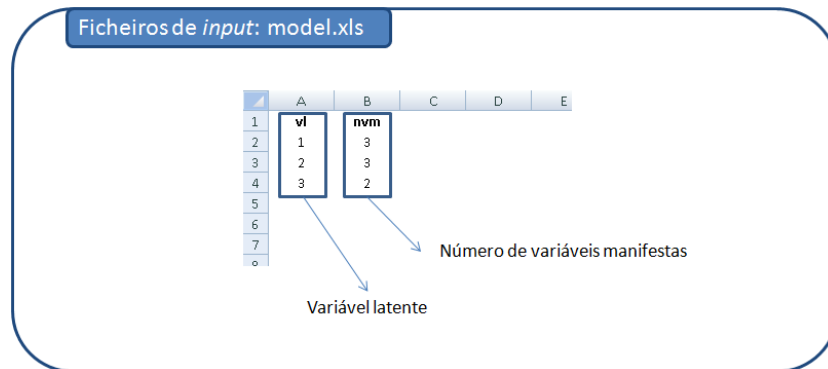


Figura 3.5: Informação de Arranque do Algoritmo: o ficheiro “model.xls”
(respectivamente B1 e B2). A estrutura deste ficheiro é:

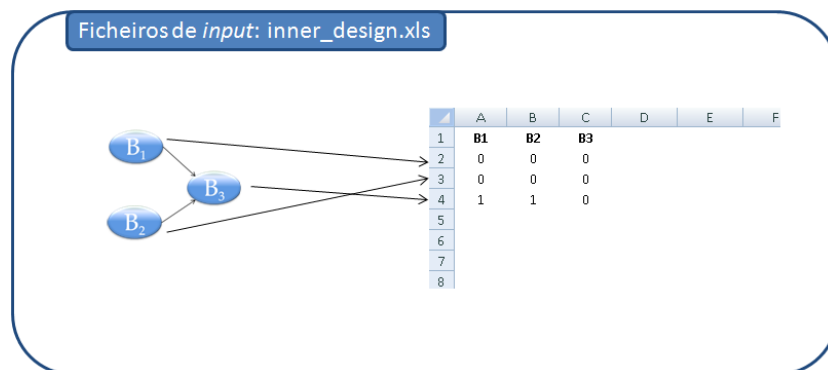


Figura 3.6: Informação de Arranque do Algoritmo: o ficheiro “inner_design.xls”

3.4 *Outputs* Desenvolvidos

Nesta secção serão apresentados os *outputs* apresentados pelo programa. Estes serão exportados em formato Excel e visualizados em diversas folhas de

dados.

Os *outputs* apresentados no ficheiro acima citado são:

- **Dados padronizados:** esta folha de dados (“Dados_std”) contém os dados iniciais padronizados;
- **Critério de paragem do algoritmo:** na folha “Stop_Criterion_Changes” estão apresentados os pesos externos estimados nos diversos *steps* que levaram à convergência dos mesmos;
- **Variáveis latentes:** na folha de dados “Latent_Variables_Scores” estão apresentados os valores (*scores*) das variáveis latentes para cada observação;
- **Loadings do modelo de medida:** na folha de dados “Outer_Loadings” estão apresentados os *loadings* associados às variáveis manifestas e calculados como correlação entre a variável latente associada ao bloco e as variáveis manifestas do mesmo;
- **Coefficientes das relações entre as variáveis latentes:** na folha “Path_Coefficients” estão apresentados os coeficientes associados às relações existentes entre as variáveis latentes (modelo estrutural). Estes coeficientes foram calculados através de uma regressão entre a variável latente dependente e as variáveis latentes ligadas à dependente;

- **Pesos externos:** na folha de dados “Outer_Weights” estão apresentados os pesos externos. Estes pesos foram utilizados para o critério de paragem do algoritmo e são calculados através de uma regressão das variáveis manifestas de um determinado bloco sobre a variável latente daquele bloco.
- **Médias das estimativas:** na folha “Bootstrap_Means” estão apresentadas as médias dos parâmetros estimados calculadas com base nas iterações da técnica de *Bootstrap*;
- **Desvios padrão das estimativas:** nesta folha de dados estão apresentados os desvios padrão dos parâmetros estimados calculados com base nas iterações da técnica de *Bootstrap*. O nome da folha é “Bootstrap_Standard_Deviation”;
- **Valores de teste:** na folha “Test_Value” estão apresentados os valores de teste associados aos parâmetros estimados. O valor de teste é calculado como ratio entre o valor médio e o desvio padrão de *Bootstrap*;
- **Correlações entre as variáveis latentes:** na folha “LVs_Correlations” estão apresentadas as correlações entre todas as variáveis latentes;
- **R^2 das regressões do modelo estrutural:** a folha de dados “In-

ner_RSquare” contém os coeficientes R^2 de cada regressão presente no modelo estrutural;

- **Comunalidade:** na folha de dados “Communality” está apresentado o índice de comunalidade. Este é utilizado para medir a qualidade do modelo externo relativa a cada bloco de variáveis manifestas. No presente trabalho a comunalidade corresponde ao AVE devido à estandardização das variáveis;
- **Redundância:** a folha de dados “Redundancy” contém o índice de redundância. Este índice mede a qualidade do modelo estrutural para cada bloco endógeno, tendo em conta o modelo de medida;
- **Alpha de Cronbach:** na folha de dados “Alpha” estão apresentados os resultados deste índice para cada bloco de variáveis manifestas. Este é utilizado para averiguar a unidimensionalidade do bloco de variáveis manifestas;
- **Rho de Dillon-Goldstein:** na folha de dados “Rho” estão apresentados os resultados deste índice que, tal como o índice apresentado anteriormente, é utilizado para o controlo da unidimensionalidade do bloco de variáveis manifestas;
- **Goodness of Fit:** a folha de dados “GoF” contém o índice de bondade

de ajustamento.

Capítulo 4

Aplicação a Dados Reais

4.1 Introdução

Este capítulo tem o objectivo de, numa primeira fase, apresentar os resultados obtidos com o software desenvolvido e, numa segunda fase, comparar esses resultados com os de um software disponível no mercado - o SmartPLS.

4.2 Os Dados Iniciais

Os dados utilizados para análise e comparação dos resultados são referentes a 250 inquéritos de satisfação com os operadores de rede móvel e estão disponíveis no SmartPLS.

Os dados utilizados apresentam 24 variáveis manifestas¹ medidas numa escala de 1 a 10 (em que 1 corresponde a um ponto de vista negativo e 10 a um positivo). Estas variáveis são utilizadas para a estimação das seguintes variáveis latentes: Expectativas, Imagem, Qualidade Apercebida, Valor Apercebido, Satisfação do cliente, Reclamações e Lealdade do cliente. Os nomes das variáveis apresentadas nos resultados são os nomes originais do inquérito, isto é, *Expectation*, *Image*, *Quality*, *Value*, *Satisfaction*, *Complaints* e *Loyalty*.

A estrutura do inquérito apresentado dá origem ao seguinte modelo:

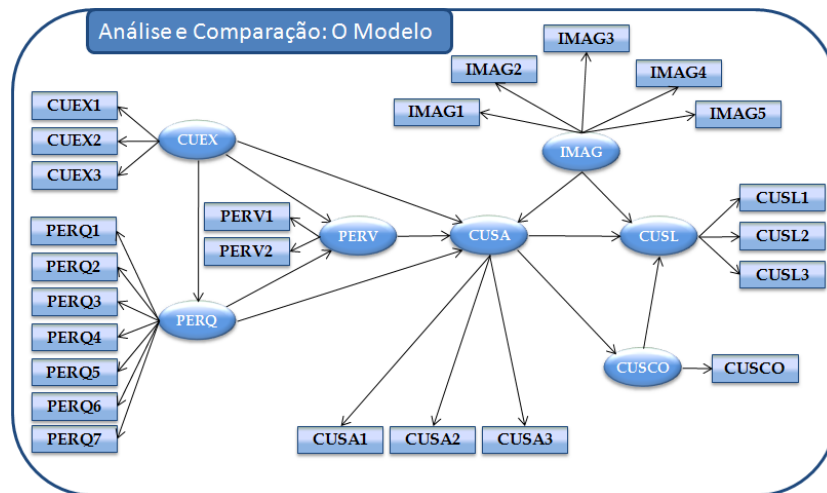


Figura 4.1: Modelo de Satisfação com os Operadores de Rede Móvel

O modelo da figura 4.1 compõe-se de dois sub-modelos: o modelo estrutural (4.2) e o modelo de medida (4.3).

¹Consultar o Anexo 2 para uma visão completa das perguntas.

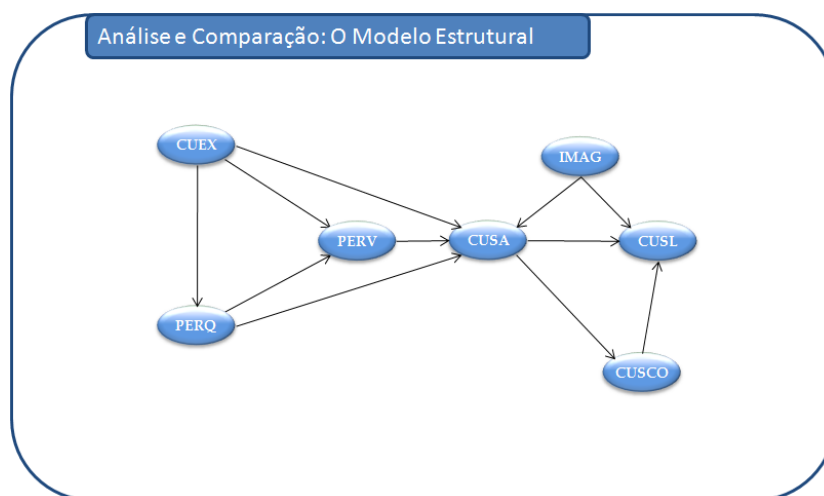


Figura 4.2: Modelo Estrutural

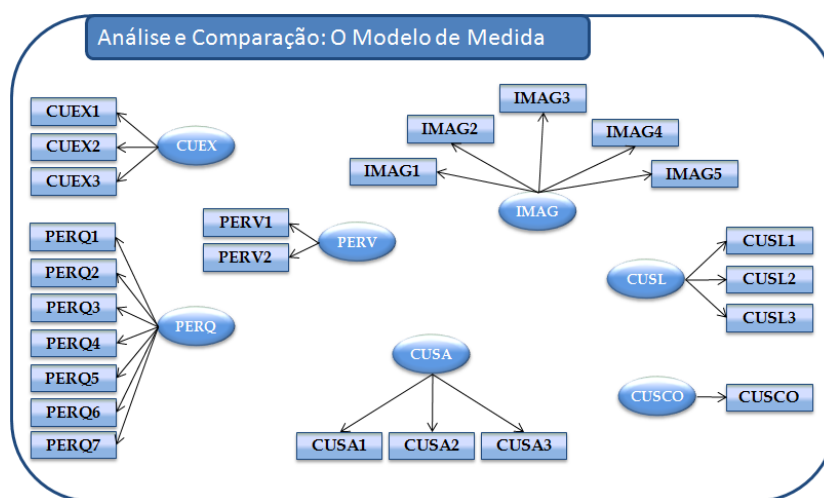


Figura 4.3: Modelo de Medida

4.3 Os Resultados

Conforme referido anteriormente, o software desenvolvido produz resultados em Excel organizando cada tipo de resultado numa folha de dados diferente

(ver figura 4.4).



Figura 4.4: O Ficheiro de *Output*

A primeira tabela de resultados apresenta os *loadings* do modelo de medida, calculados como correlação entre a variável latente e as variáveis manifestas a esta associadas:

	Expectation	Image	Quality	Value	Satisfaction	Complaints	Loyalty
CUEX1	0,7783	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
CUEX2	0,6598	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
CUEX3	0,6304	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
IMAG1	0,0000	0,7445	0,0000	0,0000	0,0000	0,0000	0,0000
IMAG2	0,0000	0,5851	0,0000	0,0000	0,0000	0,0000	0,0000
IMAG3	0,0000	0,5761	0,0000	0,0000	0,0000	0,0000	0,0000
IMAG4	0,0000	0,7734	0,0000	0,0000	0,0000	0,0000	0,0000
IMAG5	0,0000	0,7514	0,0000	0,0000	0,0000	0,0000	0,0000
PERQ1	0,0000	0,0000	0,8036	0,0000	0,0000	0,0000	0,0000
PERQ2	0,0000	0,0000	0,6375	0,0000	0,0000	0,0000	0,0000
PERQ3	0,0000	0,0000	0,7835	0,0000	0,0000	0,0000	0,0000
PERQ4	0,0000	0,0000	0,7689	0,0000	0,0000	0,0000	0,0000
PERQ5	0,0000	0,0000	0,7557	0,0000	0,0000	0,0000	0,0000
PERQ6	0,0000	0,0000	0,7750	0,0000	0,0000	0,0000	0,0000
PERQ7	0,0000	0,0000	0,7795	0,0000	0,0000	0,0000	0,0000
PERV1	0,0000	0,0000	0,0000	0,9045	0,0000	0,0000	0,0000
PERV2	0,0000	0,0000	0,0000	0,9378	0,0000	0,0000	0,0000
CUSA1	0,0000	0,0000	0,0000	0,0000	0,7989	0,0000	0,0000
CUSA2	0,0000	0,0000	0,0000	0,0000	0,8463	0,0000	0,0000
CUSA3	0,0000	0,0000	0,0000	0,0000	0,8519	0,0000	0,0000
CUSCO	0,0000	0,0000	0,0000	0,0000	0,0000	1,0000	0,0000
CUSL1	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,8139
CUSL2	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,2194
CUSL3	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,9167

Figura 4.5: *Loadings* do Modelo de Medida

Seguidamente são apresentados os *Path Coefficients* que representam os coeficientes (ou pesos) das relações existentes entre as variáveis latentes do modelo estrutural:

	Expectation	Image	Quality	Value	Satisfaction	Complaints	Loyalty
Expectation	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Image	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Quality	0,5587	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Value	0,0544	0,0000	0,5551	0,0000	0,0000	0,0000	0,0000
Satisfaction	0,0620	0,1820	0,5120	0,1912	0,0000	0,0000	0,0000
Complaints	0,0000	0,0000	0,0000	0,0000	0,5261	0,0000	0,0000
Loyalty	0,0000	0,1979	0,0000	0,0000	0,4822	0,0701	0,0000

Figura 4.6: Coeficientes das relações entre as variáveis latentes

Após os *Path Coefficients* são apresentados os pesos externos (ou *Outer Weights*) calculados como regressão das variáveis manifestas de um determinado bloco sobre a variável latente associada:

	Expectation	Image	Quality	Value	Satisfaction	Complaints	Loyalty
CUEX1	0,5363	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
CUEX2	0,4348	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
CUEX3	0,4691	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
IMAG1	0,0000	0,3011	0,0000	0,0000	0,0000	0,0000	0,0000
IMAG2	0,0000	0,2408	0,0000	0,0000	0,0000	0,0000	0,0000
IMAG3	0,0000	0,2134	0,0000	0,0000	0,0000	0,0000	0,0000
IMAG4	0,0000	0,3360	0,0000	0,0000	0,0000	0,0000	0,0000
IMAG5	0,0000	0,3356	0,0000	0,0000	0,0000	0,0000	0,0000
PERQ1	0,0000	0,0000	0,2142	0,0000	0,0000	0,0000	0,0000
PERQ2	0,0000	0,0000	0,1436	0,0000	0,0000	0,0000	0,0000
PERQ3	0,0000	0,0000	0,1994	0,0000	0,0000	0,0000	0,0000
PERQ4	0,0000	0,0000	0,1777	0,0000	0,0000	0,0000	0,0000
PERQ5	0,0000	0,0000	0,1807	0,0000	0,0000	0,0000	0,0000
PERQ6	0,0000	0,0000	0,1802	0,0000	0,0000	0,0000	0,0000
PERQ7	0,0000	0,0000	0,2145	0,0000	0,0000	0,0000	0,0000
PERV1	0,0000	0,0000	0,0000	0,4862	0,0000	0,0000	0,0000
PERV2	0,0000	0,0000	0,0000	0,5974	0,0000	0,0000	0,0000
CUSA1	0,0000	0,0000	0,0000	0,0000	0,3770	0,0000	0,0000
CUSA2	0,0000	0,0000	0,0000	0,0000	0,3817	0,0000	0,0000
CUSA3	0,0000	0,0000	0,0000	0,0000	0,4411	0,0000	0,0000
CUSCO	0,0000	0,0000	0,0000	0,0000	0,0000	1,0000	0,0000
CUSL1	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,4507
CUSL2	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,1316
CUSL3	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,6592

Figura 4.7: Pesos Externos

Na seguinte tabela estão apresentados os coeficientes de correlação entre as variáveis latentes estimadas pelo algoritmo:

	Expectation	Image	Quality	Value	Satisfaction	Complaints	Loyalty
Expectation	1,0000	0,4991	0,5587	0,3646	0,5086	0,2548	0,3765
Image	0,4991	1,0000	0,7482	0,5081	0,6931	0,4777	0,5656
Quality	0,5587	0,7482	1,0000	0,5855	0,7948	0,5315	0,5376
Value	0,3646	0,5081	0,5855	1,0000	0,6061	0,3547	0,5296
Satisfaction	0,5086	0,6931	0,7948	0,6061	1,0000	0,5261	0,6562
Complaints	0,2548	0,4777	0,5315	0,3547	0,5261	1,0000	0,4183
Loyalty	0,3765	0,5656	0,5376	0,5296	0,6562	0,4183	1,0000

Figura 4.8: Correlações entre as Variáveis Latentes

Após apresentação dos resultados, são apresentadas algumas medidas de validade e fiabilidade dos modelos (estrutural e de medida). Os indicadores produzidos pelo software são Comunalidade, Redundância, α de Cronbach, ρ de Dillon - Goldstein e o GOF (índice de bondade de ajustamento) e estão apresentados na figura 4.9².

	R²	Comunalidade	Redundância	α de Cronbach	ρ de Dillon-Goldstein	GOF
Expectation	0,0000	0,4795	0,0000	0,4519	0,7317	0,4407
Image	0,0000	0,4782	0,0000	0,7228	0,8193	
Quality	0,3121	0,5766	0,1800	0,8770	0,9050	
Value	0,3449	0,8488	0,2927	0,8236	0,9190	
Satisfaction	0,6805	0,6934	0,4718	0,7792	0,8718	
Complaints	0,2768	1,0000	0,2768	1,0000	1,0000	
Loyalty	0,4577	0,5170	0,2366	0,4724	0,7288	

Figura 4.9: Medidas de validade e fiabilidade dos modelos

²Os *output* apresentados, embora estejam em folhas de dados diferentes, foram agrupados na figura 4.9 na mesma tabela, com o objectivo de facilitar a leitura dos mesmos.

4.4 Processo de Comparação

O processo de comparação tem como objectivo validar as estimações efectuadas pelo algoritmo desenvolvido. Para tal foi seleccionado um dos softwares disponíveis no mercado e, com base nos mesmos dados, foram executados os algoritmos. Nos seguintes parágrafos é apresentado o software utilizado para a comparação e são comparados os principais resultados.

4.4.1 O SmartPLS

De entre os inúmeros softwares disponíveis, decidiu-se utilizar o SmartPLS 2.0.M3 ³. O SmartPLS (Ringle et al. 2005) é um software baseado na tecnologia Java e funciona em todos os sistemas operativos.

O modelo estrutural pode ser especificado através de um interface gráfico muito avançado e os resultados são disponíveis em diversas fontes: HTML, Latex e Excel. Para este último a criação de um ficheiro com os resultados não é directa, na medida em que, para obter os resultados no ficheiro, é necessário copiá-los do SmartPLS para um ficheiro Excel. Além da execução do modelo, o SmartPLS, apresenta também diversas técnicas de re-amostragem.

³O SmartPLS está disponível gratuitamente em <http://www.smartpls.de>.

4.4.2 O SmartPLS: os Resultados

Neste secção é efectuada uma comparação entre os resultados obtidos através do software desenvolvido e o SmartPLS.

Para tornar possível a comparação dos resultados é necessário utilizar as mesmas condições em ambos os softwares. A saber:

- As relações entre as variáveis latentes e as manifestas são de tipo reflectivo;
- Para a estimação dos pesos internos (*Path Coefficients*) foi utilizado o esquema do Centróide;
- O critério de paragem do algoritmo tem como objectivo o valor $1.0e-5$.

Sucessivamente são apresentados parte dos resultados obtidos através do SmartPLS.

4.4.3 Comparação dos Resultados

Após análise dos resultados foram analisadas as diferenças entre os dois softwares:

- *Loadings* do modelo de medida: os *loadings* não apresentam diferenças;

	Expectation	Image	Quality	Value	Satisfaction	Complaints	Loyalty
CUEX1	0,7783	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
CUEX2	0,6598	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
CUEX3	0,6304	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
IMAG1	0,0000	0,7445	0,0000	0,0000	0,0000	0,0000	0,0000
IMAG2	0,0000	0,5851	0,0000	0,0000	0,0000	0,0000	0,0000
IMAG3	0,0000	0,5761	0,0000	0,0000	0,0000	0,0000	0,0000
IMAG4	0,0000	0,7734	0,0000	0,0000	0,0000	0,0000	0,0000
IMAG5	0,0000	0,7514	0,0000	0,0000	0,0000	0,0000	0,0000
PERQ1	0,0000	0,0000	0,8036	0,0000	0,0000	0,0000	0,0000
PERQ2	0,0000	0,0000	0,6375	0,0000	0,0000	0,0000	0,0000
PERQ3	0,0000	0,0000	0,7835	0,0000	0,0000	0,0000	0,0000
PERQ4	0,0000	0,0000	0,7689	0,0000	0,0000	0,0000	0,0000
PERQ5	0,0000	0,0000	0,7557	0,0000	0,0000	0,0000	0,0000
PERQ6	0,0000	0,0000	0,7750	0,0000	0,0000	0,0000	0,0000
PERQ7	0,0000	0,0000	0,7795	0,0000	0,0000	0,0000	0,0000
PERV1	0,0000	0,0000	0,0000	0,9045	0,0000	0,0000	0,0000
PERV2	0,0000	0,0000	0,0000	0,9378	0,0000	0,0000	0,0000
CUSA1	0,0000	0,0000	0,0000	0,0000	0,7989	0,0000	0,0000
CUSA2	0,0000	0,0000	0,0000	0,0000	0,8463	0,0000	0,0000
CUSA3	0,0000	0,0000	0,0000	0,0000	0,8519	0,0000	0,0000
CUSCO	0,0000	0,0000	0,0000	0,0000	0,0000	1,0000	0,0000
CUSL1	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,8139
CUSL2	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,2194
CUSL3	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,9167

Figura 4.10: SmartPLS: *Loadings* do Modelo de Medida

	Expectation	Image	Quality	Value	Satisfaction	Complaints	Loyalty
Expectation	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Image	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Quality	0,5587	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Value	0,0544	0,0000	0,5551	0,0000	0,0000	0,0000	0,0000
Satisfaction	0,0620	0,1820	0,5120	0,1912	0,0000	0,0000	0,0000
Complaints	0,0000	0,0000	0,0000	0,0000	0,5261	0,0000	0,0000
Loyalty	0,0000	0,1979	0,0000	0,0000	0,4822	0,0701	0,0000

Figura 4.11: SmartPLS: Coeficientes das relações entre as variáveis latentes

- Pesos internos (ou *Path Coefficients*): os *Path Coefficients* apresentam diferenças nas relações das variáveis latentes *Quality*, *Value* e *Satisfaction* com a variável latente *Expectation*. As diferenças são, respectivamente, 0,000006, $-0,000033$ e 0,000032;
- Pesos externos (ou *Outer Weights*): neste caso existem duas diferenças nas relações da variável latente *Expectation* com as variáveis manifestas CUEX2 (0,000031) e CUEX3 ($-0,000022$);

	Expectation	Image	Quality	Value	Satisfaction	Complaints	Loyalty
CUEX1	0,5363	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
CUEX2	0,4348	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
CUEX3	0,4691	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
IMAG1	0,0000	0,3011	0,0000	0,0000	0,0000	0,0000	0,0000
IMAG2	0,0000	0,2408	0,0000	0,0000	0,0000	0,0000	0,0000
IMAG3	0,0000	0,2134	0,0000	0,0000	0,0000	0,0000	0,0000
IMAG4	0,0000	0,3360	0,0000	0,0000	0,0000	0,0000	0,0000
IMAG5	0,0000	0,3356	0,0000	0,0000	0,0000	0,0000	0,0000
PERQ1	0,0000	0,0000	0,2142	0,0000	0,0000	0,0000	0,0000
PERQ2	0,0000	0,0000	0,1436	0,0000	0,0000	0,0000	0,0000
PERQ3	0,0000	0,0000	0,1994	0,0000	0,0000	0,0000	0,0000
PERQ4	0,0000	0,0000	0,1777	0,0000	0,0000	0,0000	0,0000
PERQ5	0,0000	0,0000	0,1807	0,0000	0,0000	0,0000	0,0000
PERQ6	0,0000	0,0000	0,1802	0,0000	0,0000	0,0000	0,0000
PERQ7	0,0000	0,0000	0,2145	0,0000	0,0000	0,0000	0,0000
PERV1	0,0000	0,0000	0,0000	0,4862	0,0000	0,0000	0,0000
PERV2	0,0000	0,0000	0,0000	0,5974	0,0000	0,0000	0,0000
CUSA1	0,0000	0,0000	0,0000	0,0000	0,3770	0,0000	0,0000
CUSA2	0,0000	0,0000	0,0000	0,0000	0,3817	0,0000	0,0000
CUSA3	0,0000	0,0000	0,0000	0,0000	0,4411	0,0000	0,0000
CUSCO	0,0000	0,0000	0,0000	0,0000	0,0000	1,0000	0,0000
CUSL1	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,4507
CUSL2	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,1316
CUSL3	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,6592

Figura 4.12: SmartPLS: Pesos Externos

	Expectation	Image	Quality	Value	Satisfaction	Complaints	Loyalty
Expectation	1,0000	0,4991	0,5587	0,3646	0,5086	0,2548	0,3765
Image	0,4991	1,0000	0,7482	0,5081	0,6931	0,4777	0,5656
Quality	0,5587	0,7482	1,0000	0,5855	0,7948	0,5315	0,5376
Value	0,3646	0,5081	0,5855	1,0000	0,6061	0,3547	0,5296
Satisfaction	0,5086	0,6931	0,7948	0,6061	1,0000	0,5261	0,6562
Complaints	0,2548	0,4777	0,5315	0,3547	0,5261	1,0000	0,4183
Loyalty	0,3765	0,5656	0,5376	0,5296	0,6562	0,4183	1,0000

Figura 4.13: SmartPLS: Correlações entre as Variáveis Latentes

	R Square	Communality	Redundancy	Cronbachs Alpha	Composite Reliability
Expectation	0,0000	0,4795	0,0000	0,4519	0,7326
Image	0,0000	0,4782	0,0000	0,7228	0,8185
Quality	0,3121	0,5766	0,1790	0,8770	0,9047
Value	0,3449	0,8488	0,0311	0,8236	0,9182
Satisfaction	0,6805	0,6934	0,0411	0,7792	0,8715
Complaints	0,2768	1,0000	0,2768	1,0000	1,0000
Loyalty	0,4577	0,5170	0,0269	0,4724	0,7241

Figura 4.14: SmartPLS: Medidas de validade e fiabilidade dos modelos

- Correlação entre as variáveis latentes: nas correlações encontraram-se algumas diferenças na ordem de $1,00e^{-05}$;
- Medidas de validade e fiabilidade dos modelos: na análise destes indicadores encontraram-se algumas diferenças para a Redundância (na ordem das décimas) e para o ρ de Dillon - Goldstein (na ordem das milésimas).

Capítulo 5

Síntese e Conclusão

O projecto apresentado integra-se no domínio da estatística computacional. O elemento central deste trabalho é, concretamente, o desenvolvimento de um software estatístico que traz a possibilidade de executar uma metodologia(PLS-PM) que não está implementada no ambiente SAS.

Os softwares representam, hoje em dia, um importante instrumento de validação e comparação entre técnicas diferentes e, em geral, um recurso para investigação empírica sobre fenómenos reais.

Embora o software desenvolvido resulte numa ferramenta flexível (podendo ser utilizado para fins profissionais ou académicos), apresenta algumas margens de melhoria. Concretamente, de entre as perspectivas ligadas à sua utilização, destacam-se:

- o desenvolvimento do método Formativo e MIMIC para a estimação dos pesos internos e de outros esquemas para a estimação dos pesos externos (*Factor Weighting Scheme* e *Path Weighting Scheme*);
- a optimização de alguns módulos implementados, como por exemplo, o módulo relativo ao *Bootstrap* e o módulo central, no que diz respeito às medidas de qualidade e fiabilidade do modelo;
- a criação de uma interface gráfica que permita a criação gráfica do modelo e a visualização directa dos *outputs*;
- o desenvolvimento de outras medidas de qualidade e fiabilidade do modelo, como por exemplo, implementação do *Blindfolding* como técnica de *cross-validation* para comunalidade e redundância;
- a implementação do procedimento *Jack-knife* para o cálculo da significância dos parâmetros estimados.

Conforme referido ao longo da dissertação, o software criado disponibiliza, além dos *outputs* de base do PLS-PM, uma nova técnica de simulação que permite adequar o cálculo da significância às exigências do investigador.

Este projecto representa um ponto de partida para desenvolvimentos futuros devido à circunstância de ter sido livremente disponibilizado para a comunidade científica.

Anexo 1

PLS-PM: O Algoritmo SAS

```
1  libname import 'C:\PLS-PM\import';
2  libname export 'C:\PLS-PM\export';
3  libname results 'C:\PLS-PM\results';
4  libname support 'C:\PLS-PM\support';
5
6  proc import datafile='C:\PLS-PM\import\model.xls'
7              out = import.model
8              replace;
9  run;
10 proc import datafile='C:\PLS-PM\import\inner_design.xls'
11           out = import.inner_design
12           replace;
13 run;
14 proc import datafile='C:\PLS-PM\import\dados.xls'
15           out = import.dados
16           replace;
```

```

17  run;

18  proc import datafile='C:\PLS-PM\import\sim_n.xls'

19          out = import.simul

20          replace;

21  run;

22

23  proc iml;

24

25  /*****

26

27  /* MODULE: Data Generation */

28  start genData(n,vars) global (dados);

29

30      dados = J(n,vars,.);

31

32      do i = 1 to vars;

33          LM = J(n,1,.);

34          CALL randgen(LM,'BETA',4,4);

35          LM = round(LM * 9 + 1 , 1);

36          dados[,i]=LM;

37      end;

38

39      print 'MESSAGE: Data Generation ... DONE';

40

41  finish genData;

```



```

42  /*****
43
44  /* MODULE: Data standardization */
45  start stdize (x,i2,flag) global(dados_std);
46
47      mean = x[:,1:i2]; /* mean calculation for variables between 1 and "i2"*/
48      x[:,1:i2] = x[:,1:i2]-repeat(mean,nrow(x),1); /* centering data */
49      ss = x[##,1:i2] ; /* sum of squares for variables between 1 and "i2" */
50      std = sqrt(ss / (nrow(x)-1)); /* standard deviation calculation */
51      x[:,1:i2] = x[:,1:i2] * diag(1/std); /* centered data normalization */
52      dados_std = x;
53      return (dados_std);
54
55      print 'MESSAGE: Data Standardization ... DONE ';
56
57  finish stdize;
58  /*****
59
60  /* MODULE: Algorithm initialization */
61  start inlize(n,x,model) global(LVs);
62
63      LVs = J(n,nrow(model),.);
64      LV = J(n,1,.);
65      a = 0;
66

```

```

67     do i = 1 to nrow(model);
68         a = sum(model[1:i,2]);
69         b = model[i,2];
70         c = 1 + a - b;
71         LV = x[,c];
72         LVs[,i]=LV;
73     end;
74     return (LVs);
75     print 'MESSAGE: Latent Variables with weights [1,0,...,0] ... DONE';
76
77 finish inlize;
78 /*****
79
80 /* MODULE: Structural model estimation (INNER ESTIMATION) */
81
82 start inner_est(x,model,imtx) global(LVs_Inn, dados_std);
83
84     im2 = imtx + t(imtx);
85     print imtx;
86     corr_im = sign(corr(x));
87     corr_im2 = corr_im # im2;
88     print corr_im2;
89     LVs_Inn = J(nrow(x),nrow(imtx),0);
90     do i = 1 to ncol(imtx);
91         do j = 1 to nrow(imtx);

```

```

92             if im2[i,j] = 1 then do;
93                 term = x[,j] * corr_im2[i,j];
94                 LVs_Inn[i] = LVs_Inn[i] + term;
95             end;
96         end;
97     end;
98     return (LVs_Inn);
99     print 'MESSAGE: Inner Estimation ... DONE';
100
101 finish inner_est;
102 /*****
103
104 /* MODULE: Measuring model estimation (OUTER ESTIMATION) */
105
106 start outer_est(x,lv,model) global(LVs_Out, dados_std);
107
108     a = 0;
109     r_compl = J(model[+,2],nrow(model),0);
110     r = J(model[+,2],nrow(model),0);
111     r_ln = J(model[+,2],1,0);
112     do i = 1 to nrow(model);
113         do j = 1 to model[i,2];
114             a = a + 1;
115             m = x[,a];
116             n = lv[,i];

```

```

117         b = m||n;
118         r_compl = corr(b);
119         r[a,i] = r_compl[2,1];
120         r_ln[a,1] = r_compl[2,1];
121     end;
122 end;
123 LVs_Out = x * r;
124 i = nrow(model);
125 LVs_Out = stdize(LVs_Out,i,1);
126     return (LVs_Out);
127
128     print 'MESSAGE: Outer Estimation ... DONE';
129
130 finish outer_est;
131 /*****
132
133  /* MODULE: Multiple Linear Regression */
134
135  start reg (x,y,flag);
136
137      n = nrow(x);                                /* number of observations */
138      k = ncol(x);                                /* number of variables */
139      xpx = t(x) * x;                             /* cross-products */
140      xpy = t(x) * y;
141      xpxi = inv(xpx);                             /* inverse crossproducts */

```

```

142     b = xpxi * xpy;                                /* parameter estimates */
143     yhat = x * b;                                    /* predicted values */
144     resid = y - yhat;                                /* residuals */
145     sse = t(resid) * resid;                          /* sum of squared errors */
146     dfe = n - k;                                    /* degrees of freedom error */
147     mse = sse / dfe;                                /* mean squared error */
148     rmse = sqrt(mse);                               /* root mean squared error */
149     cssy=ssq(y-sum(y)/n);                          /* corrected total SS */
150     rsquare=(cssy-sse)/cssy;                        /* RSQUARE */
151     covb = xpxi # mse;                              /* covariance of estimates */
152     stdb = sqrt(vecdiag(covb));                     /* standard errors */
153     *t = b / stdb;                                  /* ttest for estimates=0 */
154     *probt = 1 - probf(t#t,1,dfe);                  /* significance probability */
155     *s = diag(1 / stdb);
156     *corrb = s * covb * s;                          /* correlation of estimates */
157     *projx = x * xpxi * t(x);                       /* hat matrix */
158     *vresid = (i(n) - projx) * mse;                 /* covariance of residuals */
159     *vpred = projx # mse;                           /* cov of predicted values */
160     *h=vecdiag(projx);                              /* hat leverage values */
161     if flag = 0 then return (b);
162     else return (rsquare);
163 finish reg;
164 /*****
165
166 /* MODULE: Outer weights calculation */

```

```

167
168 start outer_wgt(x,lv,flag) global (out_wgt, out_wgt_ln);
169
170     use import.model;
171     read all var _num_ into model;
172     b = model[1,2];
173     out_wgt = J(model[+,2],nrow(model),0);
174     out_wgt_ln = J(model[+,2],1,0);
175     a = 0;
176
177     do i = 1 to nrow(model);
178         if i = 1 then a = a + model[(i),2];
179         else a = a + model[(i-1),2];
180         inf = 1 + a - b;
181         sup = inf + model[i,2] - 1;
182         ind = x[,inf:sup];
183         dep = lv[,i];
184         out_wgt[inf:sup,i] = reg(ind,dep,0);
185         out_wgt_ln[inf:sup,1] = reg(ind,dep,0);
186     end;
187     if flag = 1 then return (out_wgt);
188     if flag = 2 then return (out_wgt_ln);
189     print 'MESSAGE: Outer weights calculation ... DONE';
190 finish outer_wgt;
191 /*****

```

```

192
193  /* MODULE: Starting weight generation */
194
195  start create_wgt(model) global (wgt);
196
197      a = 0;
198      wgt = j(model[+,2],1,0);
199      do i = 1 to nrow(model);
200          do j = 1 to model[i,2];
201              a = a + 1;
202              if j = 1 then wgt[a] = 1;
203          end;
204      end;
205      return (wgt);
206      print 'MESSAGE: Starting weights generation ... DONE';
207
208  finish create_wgt;
209  /*****
210
211  /* MODULE: Algorithm iteration [Iterative] */
212
213  start core (x,lv,scrch,model,im, method)
214      global (LVs_Out,LVs_Inn,out_wgt_ln,out_ldg, out_wgt, path_cft,inn_crl);
215
216      i = 1;

```

```

217     ss = 1;
218
219     z = 100;
220
221     out_wt = J(model[+,2],nrow(model),0);
222
223     critch = scrch;
224
225     inn_lv = lv;
226
227     do while ( z >= 1.0e-5);
228
229         i = i + 1;
230
231         out_lv = outer_est (x, inn_lv,model);
232
233         inn_lv = inner_est(out_lv,model,im);
234
235         out_wt = outer_wgt(x,inn_lv, 1);
236
237         out_wt_ln = outer_wgt(x,inn_lv, 2);
238
239         critch = critch||out_wt_ln;  /* Outer weights matrix */
240
241         ss = abs(critch[,i-1] - critch[,i]);
242
243         z = sum(ss);
244
245     end;
246
247     out_wt = outer_wgt(x,out_lv, 1);
248
249     /* Outer loadings estimation */
250
251     mat = x||out_lv;
252
253
254
255
256     r = corr(mat);
257
258     out_ldg = J(model[+,2],nrow(model),0);
259
260     a = 0;
261
262
263
264
265     do i = 1 to nrow(model);
266
267         b = i + model[+,2];

```



```

242         do j = 1 to model[i,2];
243             a = a + 1;
244             out_ldg[a,i] = r[a,b];
245         end;
246     end;
247
248     /* Path coefficients and R-squares estimation */
249     path_cft = J (nrow(model),nrow(model),0);
250     rsq_mat = J(nrow(model),1,0);
251     ind = J(nrow(x),1,0);
252     do i = 2 to nrow(im);
253         min = 0;
254         max = 0;
255         do j = 1 to (i-1);
256             if im[i,j] = 1 then
257                 do;
258                     ind = ind || out_lv[,j];
259                     if min = 0 then min = j;
260                     else max = j;
261                 end;
262             end;
263
264             if max = 0 then max = min;
265             if min <> 0 then
266                 do;

```

```

267         dep = out_lv[,i];
268         ind = ind[,2:ncol(ind)];
269         a = reg(ind,dep,0);
270         b = reg(ind,dep,1);
271         rsq_mat[i,1] = b;
272         a = t(a);
273         do l = 1 to ncol(a);
274             do m = 1 to (i-1);
275                 if im[i,m] = 1 then
276                     do;
277                         path_cft[i,m] = a[,l];
278                         im[i,m] = 0;
279                         m = i;
280                     end;
281                 end;
282             end;
283         end;
284         ind = J(nrow(x),1,0);
285     end;
286
287     create results.Path_coefficients from path_cft;
288     append from path_cft;
289
290     /* Communality, Redundancy and GOF estimation */
291     comunalidade = J(nrow(model),1,0);

```

```

292     redundancia = J(nrow(model),1,0);
293     a = 0;
294     b = model[1,2];
295     do i = 1 to nrow(model);
296         if i = 1 then a = a + model[(i),2];
297         else a = a + model[(i-1),2];
298         inf = 1 + a - b;
299         sup = inf + model[i,2] - 1;
300         a_com = out_lv[,i] || x[,inf:sup];
301         comunalidade[i,1] = (corr(a_com)[##,1]-1) / model[i,2];
302         redundancia[i,1] = comunalidade[i,1] * rsq_mat[i,1];
303     end;
304     comunalidade_med = comunalidade[:];
305     rsq_mat_med = rsq_mat[:];
306     GOF = sqrt(comunalidade_med * rsq_mat_med);
307
308
309 /* Composite Reliability: Cronbach's Alpha and Dillon-Goldstein's Rho*/
310     alpha = J(nrow(model),1,0);
311     rho = J(nrow(model),1,0);
312     a = 0;
313     b = model[1,2];
314     do i = 1 to nrow(model);
315         if i = 1 then a = a + model[(i),2];
316         else a = a + model[(i-1),2];

```

```

317         inf = 1 + a - b;
318         sup = inf + model[i,2] - 1;
319         cor_mat = corr(x[,inf:sup]);
320         if nrow(cor_mat) > 1 then alpha[i,1] =
321             ((cor_mat[+]- (nrow(cor_mat)))) / (cor_mat[+])) *
322             (nrow(cor_mat) / (nrow(cor_mat)-1));
323
324         else alpha[i,1] = 1;
325         *print cor_mat alpha;
326
327         use import.dados;
328         read all var _num_ into xx;
329         mat = xx[,inf:sup];
330         n=nrow(mat);
331         r=ncol(mat);
332         * Generate the covariance and correlation matrices of X;
333         xbar = j(n)*mat/n;           * Each row has column means, ;
334         xcent = mat-xbar;           * Center covariates at covariate means;
335         covar = xcent'*xcent/(n-1); * COVARIANCE MATRIX of columns ;
336         * Next, find the data normalized so that each variable ;
337         *   has mean zero and variance one;
338         xstdev = sqrt(diag(covar));  * Diagonal matrix with stdevs of cols;
339         xnorm = xcent*inv(xstdev);   * NORMALIZED covariate columns;
340         corr = xnorm'*xnorm/(n-1);
341         * Invoke a SAS routine that store the EIGENVALUES;

```

```
342     CALL eigen(egvals,egvecs, corr);
343     y = xnorm*egvecs;
344     y = y[,1];
345     mat_compl = y||mat;
346     mat_cor = corr(mat_compl)[2:r+1];
347     num_rho = (mat_cor[+])##2;
348     sq_mat_cor = mat_cor##2;
349     sq_mat_cor2 = 1-sq_mat_cor;
350     den_rho = num_rho + sq_mat_cor2[+];
351     rho[i,1] = num_rho /den_rho;
352 end;
353
354 inn_crl = corr(out_lv);           /* Inner correlation calculation*/
355
356 if method = "Base" then
357 do;
358     create results.Stop_criterion_changes from critch;
359     append from critch;
360
361     create results.Latent_variables_scores from out_lv;
362     append from out_lv;
363
364     create results.Outer_loadings from out_ldg;
365     append from out_ldg;
366
```

```

367         create results.Outer_weights from out_wt;
368         append from out_wt;
369
370         create results.dados_std from x;
371         append from x;
372
373         create results.LVs_correlations from inn_crl;
374         append from inn_crl;
375
376         create results.Inner_rsquare from rsq_mat;
377         append from rsq_mat;
378
379         create results.Communality from comunalidade;
380         append from comunalidade;
381
382         create results.Redundancy from redundancia;
383         append from redundancia;
384
385         create results.Gof from GOF;
386         append from GOF;
387
388         create results.Alpha from alpha;
389         append from alpha;
390
391         create results.Rho from rho;

```

```

392         append from rho;

393

394         print "Starting Standardized Data" ,, x;
395         print "Stop Criterion Changes" ,, critch;
396         print "Latent Variables" ,, out_lv;
397         print "Outer Loadings" ,, out_ldg;
398         print "Path Coefficients" ,, path_cft;
399         print "Outer Weights" ,, out_wt;
400         print "LV's Correlations" ,, inn_crl;
401         print "Inner R-square" ,, rsq_mat;
402         print "Communality" ,, comunalidade;
403         print "Redundancy" ,, redundancia;
404         print "Cronbach's Alpha" ,, alpha;
405         print "Dillon-Goldstein's Rho" ,, rho;
406         print "GoF (Goodness-of-Fit)" ,, GOF;

407     end;

408 finish core;

409 /*****
410
411  /* MODULE: Bootstrap method [Parameters significance]*/
412
413  start bootstrap(x,times,model_sp)
414
415      global(inn_dsg,testval,dados_std,LVs,LVs_Out,LVs_Inn,out_wgt_ln,out_ldg,
416
417          out_wgt, path_cft,wgt,med,std);

416      meth = ("Bootstrap");

```

```

417     sample = J(nrow(x),ncol(x),0);      /* Sample matrix */
418     soma_x = J(nrow(model_sp),nrow(model_sp),0);
419     soma_x2 = J(nrow(model_sp),nrow(model_sp),0);
420     med = soma_x;
421     std = soma_x;
422     c = soma_x;
423     do t = 1 to times;
424         do i = 1 to nrow(x);
425             val = ROUND(1 + ((nrow(x)-1) * RANUNI(0)));
426             sample[i,] = x[val,];
427         end;
428         use import.inner_design;
429         read all var _num_ into inn_dsg;
430         /* Model estimation*/
431         Mvars = model_sp[+,2];
432         n = nrow(sample);
433         *   Standardizing data;
434         dados_in = stdize(sample,Mvars,1);
435
436         *   Initializing algorithm;
437         LVs = inlize(n,dados_in,model_sp);
438
439         *   Inner estimation;
440         print inn_dsg;
441         st_lvs_inn = inner_est(LVs,model_sp,inn_dsg);

```



```

442
443      *   Starting weights generation;
444      wgt = create_wgt (model_sp);
445
446      *   Algorithm iteration [Iterative];
447      CALL core(dados_in,st_lvs_inn, wgt, model_sp, inn_dsg, meth);
448      *   Mean and Standard Deviation Calculation;
449      use results.Path_coefficients;
450      read all var _num_ into path_cft;
451      close results.Path_coefficients;
452      soma_x = soma_x + path_cft;
453      soma_x2 = soma_x2 + ((path_cft)**2);
454  end;
455      med = soma_x / times;
456      a = soma_x2 / (times - 1);
457      b = (med**2);
458      c = a-b;
459      std = sqrt(c);
460      * Do statements that prevent "Division by 0";
461      do i = 1 to nrow(model_sp);
462          do j = 1 to nrow(model_sp);
463              if std[i,j]= 0 then std[i,j]= 10000000;
464          end;
465      end;
466      testval = med / sqrt(std);

```

```

467         *print a b c std;
468         close import.inner_design;
469     finish bootstrap;
470
471     /*****
472
473     /* MODULE: Bootstrap method Simulation Program (all datas in one Excel)*/
474
475     start bootstrap_sim(x,model_sp)
476         global(testval,dados_std,LVs,LVs_Out,LVs_Inn,out_wgt_ln,out_ldg, out_wgt,
477             path_cft,wgt,med,std);
478         meth = ("Bootstrap");
479
480         sample = J(nrow(x),ncol(x),0);      /* Sample matrix */
481         soma_x = J(nrow(model_sp),nrow(model_sp),0);
482         soma_x2 = J(nrow(model_sp),nrow(model_sp),0);
483         med = soma_x;
484         std = soma_x;
485         c = soma_x;
486
487         use import.simul;
488         read all var _num_ into sim_n;
489
490         times = nrow(sim_n);
491         inf = 1;

```

```

492     sup = 0;
493     teste = J(times,2,0);
494     do t = 1 to times;
495         use import.inner_design;
496     read all var _num_ into inn_dsg;
497         /* Data selection */
498         if t > 1 then inf = inf + sim_n[t-1,1];
499         sup = sup + sim_n[t,1];
500         teste[t,1] = inf;
501         teste[t,2] = sup;
502         sample = x[inf:sup,];
503         /* Model estimation*/
504         Mvars = model_sp[+,2];
505         n = nrow(sample);
506         *   Standardizing data;
507         dados_in = stdize(sample,Mvars,1);
508         *   Initializing algorithm;
509         LVs = inlize(n,dados_in,model_sp);
510         *   Inner estimation;
511         st_lvs_inn = inner_est(LVs,model_sp,inn_dsg);
512         *   Starting weights generation;
513         wgt = create_wgt (model_sp);
514         *   Algorithm iteration [Iterative];
515         CALL core(dados_in,st_lvs_inn, wgt, model_sp, inn_dsg, meth);
516         *   Mean and Standard Deviation Calculation;

```

```

517         use results.Path_coefficients;
518         read all var _num_ into path_cft;
519         soma_x = soma_x + path_cft;
520         soma_x2 = soma_x2 + ((path_cft)##2);
521     end;
522     print teste;
523     med = soma_x / times;
524     a = soma_x2 / (times - 1);
525     b = (med##2);
526     c = a-b;
527     std = sqrt(c);
528     testval = med / sqrt(std);
529     *close results.Path_coefficients;
530     close import.inner_design;
531 finish bootstrap_sim;
532
533 /*****
534
535 /* MODULE: Main */
536
537 start main;
538     use import.model;
539     read all var _num_ into model_sp;
540     Mvars = model_sp[+,2];
541     use import.dados;

```

```

542     read all var _num_ into dados;
543     n = nrow(dados);
544
545     /* Bootstrap (SIMULATION MODE) */
546         *use import.dados_boot;
547         *read all var _num_ into dados_b;
548         *CALL bootstrap_sim(dados,model_sp);
549     /* Fim Bootstrap (SIMULATION MODE)*/
550
551     CALL bootstrap(dados,100,model_sp);
552
553     create results.Bootstrap_means from med;
554     append from med;
555     create results.Bootstrap_Standard_Deviation from std;
556     append from std;
557     create results.Test_Value from testval;
558     append from testval;
559
560     meth = ("Base");
561 *   Standardizing data                                OK;
562     d_std = stdize(dados,Mvars,1);
563     create import.dados_std from d_std;
564     append from d_std;
565     use import.dados_std;
566     read all var _num_ into dados_in;

```

```

567 *   Initializing algorithm                                OK;
568     LVs = inlize(n,dados_in,model_sp);
569     create export.LVs from LVs;
570     append from LVs;
571 *   Inner estimation          OK;
572     use import.inner_design;
573     read all var _num_ into inn_dsg;
574     LVs_I = inner_est(LVs,model_sp,inn_dsg);
575     create export.starting_LVs from LVs_I;
576     append from LVs_I;
577     use export.starting_LVs;
578     read all var _num_ into st_lvs_inn;
579 *   Starting weights generation;
580     wgt = create_wgt (model_sp);
581     create export.starting_wgt from wgt;
582     append from wgt;
583     use export.starting_wgt;
584     read all var _num_ into wgt;
585 *   Algorithm iteration [Iterative];
586     use import.inner_design;
587     read all var _num_ into inn_dsg;
588     CALL core(dados_in,st_lvs_inn, wgt, model_sp, inn_dsg, meth);
589 finish main;
590
591

```

```

592  run main;

593  quit;

594

595

596

597  %macro excel(data);

598  proc export data = results.&data

599      outfile='C:\Documents and Settings\cris\Desktop\output_PLSPM.xls'

600      dbms=EXCEL

601      replace;

602      sheet=&data;

603  run;

604  %mend;

605  %excel(dados_std);

606  %excel(Stop_criterion_changes);

607  %excel(Latent_variables_scores);

608  %excel(Outer_loadings);

609  %excel(Path_coefficients);

610  %excel(Outer_weights);

611  %excel(Bootstrap_means);

612  %excel(Bootstrap_Standard_Deviation);

613  %excel(Test_Value);

614  %excel(LVs_correlations);

615  %excel(Inner_rsquare);

616  %excel(Communality);

```

617 %excel(Redundancy);

618 %excel(Alpha);

619 %excel(Rho);

620 %excel(Gof);

Anexo 2

Inquérito de Satisfação para os Operadores de Rede Móvel (versão original)

- **Customer Expectations of the overall quality**

- Expectations for the overall quality of “your mobile phone provider” at the moment you became customer of this provider
- Expectations for “your mobile phone provider” to provide products and services to meet your personal need
- How often did you expect that things could go wrong at “your mobile phone provider”

- **Image;**

- It can be trusted in what it says and does
- It is stable and firmly established

- It has a social contribution for the society
- It is concerned with customers
- It is inovative and forward looking

- **Perceived Quality;**

- Overall perceived quality
- Technical quality of the network
- Customer service and personal advice offered
- Quality of the services you use
- Range of services and products offered
- Reliability and accuracy of the products and services provided
- Clarity and transparency of information provided

- **Perceived Value;**

- Given the quality of the products and services offered by “your mobile phone provider” how would you rate the fees and prices that you pay for them?
- Given the fees and prices that you pay for “your mobile phone provider” how would you rate the quality of the products and services offered by “your mobile phone provider”?

- **Customer Satisfaction;**

- Overall satisfaction
- Fulfillment of expectations
- How well do you think “your mobile phone provider” compares with your ideal mobile phone provider?

- **Customer Complaints;**

- You complained about “your mobile phone provider” last year.
How well, or poorly, was your most recent complaint handled

or

- You did not complain about “your mobile phone provider” last year. Imagine you have to complain to “your mobile phone provider” because of a bad quality of service or product. To what extent do you think that “your mobile phone provider” will care about your complaint?

- **Customer Loyalty;**

- If you would need to choose a new mobile phone provider how likely is it that you would choose “your provider” again?

- Let us now suppose that other mobile phone providers decide to lower their fees and prices, but “your mobile phone provider” stays at the same level as today. At which level of difference (in %) would you choose another mobile phone provider?
- If a friend or colleague asks you for advice, how likely is it that you would recommend “your mobile phone provider”?

Bibliografia

Chin, W. (1998), ‘The partial least squares approach for structural equation modeling’.

Chin, W. (2001), ‘Pls-graph user’s guide version 3.0’.

Corbetta, P. (2002), *Metodi di analisi multivariata per le scienze sociali*, Il Mulino, Bologna.

Diamantopoulos, A. & Winklhofer, H. (2001), ‘Index construction with formative indicators: an alternative to scale development’, *Journal of Marketing Research* **38**(2), 269–277.

Fornell, C., Johnson, M. D., Anderson, E. W., Cha, J. & Bryant, B. E. (2002), ‘The american customer satisfaction index: Nature, purpose, and findings’, *Journal of Marketing* **60**(4), 7–19.

Fu, J. (2006), ‘Visualpls - partial least square (pls) regression - an enhanced gui for lvpls (pls 1.8 pc) version 1.04’.

- Gherghi, M. & Lauro, C. (2004), *Appunti di Analisi dei Dati Multidimensionali*, RCE Edizioni, Napoli.
- Jiang, J. H., Berry, R. J., Siesler, H. W. & Ozaki, Y. (2002), ‘Wavelength interval selection in multicomponent spectral analysis by moving window partial least-squares regression with applications to mid-infrared and near-infrared spectroscopic data’, *Analytical Chemistry* **74**(14), 3555–3565.
- Jöreskog, K. (1970), ‘A general method for analysis of covariance structure’, *Biometrika* pp. 239–251.
- Jöreskog, K. & Wold, H. (1982), ‘The ml and pls techniques for modeling with latent variables: historical and comparative aspects’.
- Li, Y. (2005), ‘Pls-gui - graphic user interface for partial least squares (pls-pc 1.8) - version 2.0.1 beta’.
- Lohmöller, J. (1987), ‘Lvpls program manual, version 1.8.’.
- Lohmöller, J. (1989), ‘Latent variables path modeling with partial least squares’, *Physica-Verlag* .
- Ringle, C., Wende, S. & Will, A. (2005), ‘Smartpls - version 2.0’.
- SASInstitute (2004), ‘Sas/iml. 9.1 user’s guide.’.

- Sellin, N. (1989), 'Plspath - version 3.01. application manual'.
- Temme, D., Kreis, H. & Hildebrandt, L. (2005), 'Pls path modeling - a software review'.
- Tenenhaus, M., Esposito Vinzi, V., Chatelin, Y. & Lauro, C. (2005), 'Pls path modeling', *Computational Statistics and Data Analysis* **48**, 159–205.
- Test & Go (2006), 'Spad version 6.0.0'.
- Thurstone, L. L. (1931), *The theory of multiple factors*, Edwards Brothers, Ann Arbor.
- Trinchera, L. (2007), Unobserved Heterogeneity in Structural Equation Models: a new approach to latent class detection in PLS Path Modeling, PhD thesis, Università degli studi di Napoli Federico II, Napoli.
- Tukey, J. W. (1964), Causation, regression and path analysis, *in* 'Statistics and Mathematics in Biology', Hafner Publishing Company, New York.
- Vilares, M. & Coelho, P. (2005), *Satisfação e Lealdade do Cliente: Metodologias de avaliação, gestão e análise*, Escolar Editora, Lisboa.
- Wold, H. (1966), 'Estimation of principal components and related models by iterative least squares'.

- Wold, H. (1973), ‘Non-linear iterative partial least squares (nipals) modelling. some current developments’.
- Wold, H. (1975), Soft modelling by latent variables: the non-linear iterative partial least squares (nipals) approach, *in* J. Gani, ed., ‘Perspectives in Probability and Statistics’, Applied Probability Trust, London, pp. 117–142.
- Wold, H. (1982), ‘Partial least squares’.
- Wold, S., Martens, H. & Wold, H. (1983), The multivariate calibration problem in chemistry solved by the pls method, *in* A. Ruhe & B. Kågström, eds, ‘Proceedings of the Conference on Matrix Pencils’, Springer, Heidelberg.